# Automatic Extraction of Closed Contours Bounding Salient Objects: New Algorithms and Evaluation Methods

Vida Movahedi

A DISSERTATION SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

APRIL 2015

# Abstract

The problem under consideration in this dissertation is achieving *salient object segmentation* in natural images by means of probabilistic contour grouping. The goal is to extract the *simple closed contour* bounding the salient object in a given image. The method proposed here falls in the *Contour Grouping* category, searching for the optimal grouping of boundary entities to form an object contour.

Our **first contribution** is to provide both a ground truth dataset and a performance measure for empirical evaluation of salient object segmentation methods. Our *Salient Object Dataset* (SOD) provides ground truth boundaries of salient objects perceived by humans in natural images. We also psychophysically evaluated 5 distinct performance measures that have been used in the literature and showed that a measure based upon minimal contour mappings is most sensitive to shape irregularities and most consistent with human judgements. In fact, the *Contour Mapping measure* is as predictive of human judgements as human subjects are of each other.

Contour grouping methods often rely on Gestalt cues locally defined on pairs of oriented features. Accurate integration of these local cues with global cues is a challenge. A **second major contribution** of this dissertation is a novel, effective method for *combining local and global cues*.

A **third major contribution** in this dissertation is a novel method based on Principal Component Analysis for promoting *diversity* among contour hypotheses, leading to substantial improvements in grouping performance.

To further improve the performance, a *multiscale implementation* of this method has been studied. A **fourth contribution** in this dissertation is studying the effect of the multiscale prior on the performance and analysing the method for combining the results obtained in different resolutions.

Our **final contribution** is comparing the performance of univariate distribution models for local cues used by our method with the use of a *multivariate mixture model* for their joint distribution. We obtain slight improvement by the mixture models.

The proposed method has been evaluated and compared with four other state-of-the-art grouping methods, showing considerably better performance on the SOD ground truth dataset.

To Masoud, Arash, Artin, and Tina

# Acknowledgements

I would like to first thank my supervisor, Prof. James Elder. Throughout the years I have been working with him, he has always advised and guided me with diligence and patience. I have not only learned contour grouping, but also learned to do careful research and to always ask questions that would lead to better understanding and learning. I would also like to thank my supervisory committee for their advice and guidance.

I want to thank all my colleagues in Elderlab for their support as well. I specifically want to thank Bob Yhou for all his help. He always made sure that everything went smooth. And many thanks to Yaniv Morgenstern, Patrick Denis, Charles Or, Vladimir Magdin, Bob Yhou, Paria Mehrani, and the summer students who kindly participated in my psychophysical experiments and the preparation of the ground truth data.

I would like to acknowledge Francisco Estrada for his code of the Multiscale algorithm, and his input and help for getting me started in this project.

I appreciate all the memories with my friends at York University through these years. I certainly cannot name everyone, but special thanks to Tim Oleskiw, Eduardo Corral Soto, Ron Tal, Paria Mehrani and Ying Li for their valuable friendship.

And of course, many thanks to my caring husband Masoud, and my lovely children, Arash, Artin, and Tina. They were all very understanding with

my tight schedules, specially in the last few months. The “I love you, mommy”s were my sources of energy! I also like to thank my parents, my sisters and my friends for their love and support at all times.

# Table of Contents

# List of Tables

# List of Figures

# 1

# Introduction

The ultimate goal of Computer Vision is to enable a machine to see and understand an image or scene, at least as well as a human. An important step towards this goal is to partition an image into regions, each corresponding to an object or entity. This is referred to as *image segmentation* in the computer vision community. By the segmentation process, each pixel is assigned a label such that pixels with the same label share certain properties, for example belonging to the same object (or background). Segmentation is an important step towards image understanding and can enhance the performance of many applications such as object detection, object tracking, surveillance, medical imaging, etc. In this dissertation, the focus is on extracting the most salient object in the image, also referred to as *salient object segmentation* [1, 2].

Although segmentation is extremely simple and effortless for the human vision system, it is a difficult problem in Computer Vision. Segmentation is either stated as a regional labeling problem or the dual problem of boundary extraction (Figure 1.1). The cues used in a segmentation method are either region-based (e.g. color intensity of pixels), boundary-based (e.g. smoothness of boundary), or combinations of both.

Based on the operating domain of the segmentation algorithm, they can be classified into three main categories: 1) *regional segmentation methods*, which optimize the

**Figure 1.1: Salient object segmentation** - (a) A sample image (source: [3]), (b) Segmentation as regional labeling, (c) corresponding object boundary.

labeling of pixels, 2) *active contour models*, which find the optimal deformation and location of a deformable contour around an object in the image, and 3) *contour grouping methods*, which search for the optimal grouping of boundary entities (e.g. edgels, contour segments) to form an object contour.

Regional methods fail for objects with heterogenous surface properties. In the active contour models, the emphasis is on the information available on the boundary of regions; and the optimization process often starts with an initial assumed contour[1]. The method proposed in this dissertation falls in the contour grouping category. In addition to being applicable to heterogenous objects, these methods do not require initialization.

## 1.1 Contour Grouping

Segmentation can also be viewed as *perceptual grouping* of image data. Perceptual grouping is defined as "the problem of aggregating primitive image features that project from a common structure in the visual scene" [5]. The redundancy in data obtained from an image can effectively be reduced by edge detection. While discarding redundant information [6], edges provide explicit visual information that can be exploited by segmentation methods effectively. In contour grouping methods, edges (or contour fragments constructed from edges) are grouped to define the object's boundary. Cues such as proximity, smooth continuation, similarity, etc are used in the grouping

[1] Initialization-free active contour methods are an active research topic [4].

process. Many of these cues have been studied by psychologists and are known as the Gestalt principles of perceptual organization [7].

## 1.2 Problem Definition

The problem under consideration in this dissertation is achieving salient object segmentation of natural images by means of probabilistic contour grouping. The goal is to extract the *simple closed contour* bounding the most *salient object* in a natural image. Our method does not use any motion or stereo information, or user interactions. We do not make any assumptions regarding object types, shapes, color, etc., or background and lighting conditions. The method used is *probabilistic contour grouping*, i.e. searching for an optimal cycle of local oriented primitives (e.g. lines segments) forming the boundary, using probabilistic models learned from training data.

Figure 1.2(a) shows a contour grouping example. Given an image, line segments are extracted by edge detection and line approximation. The sample output consists of a (non-self-intersecting) closed polygon comprised of visible line segments (green) connected by linear interpolants (red).

Figure 1.2(b) shows a schematic of the processing pipeline in our grouping method. Given a line map obtained by edge and line detection in the image, an association graph is constructed. In this graph, a search is performed to extract plausible closed contours. These closed contours are ranked for selection as output.

## 1.3 Summary of contributions

Our contributions in solving the salient object segmentation problem using a probabilistic contour grouping method can be summarized as follows:

1. A preliminary tool in our method is an evaluation measure for measuring how

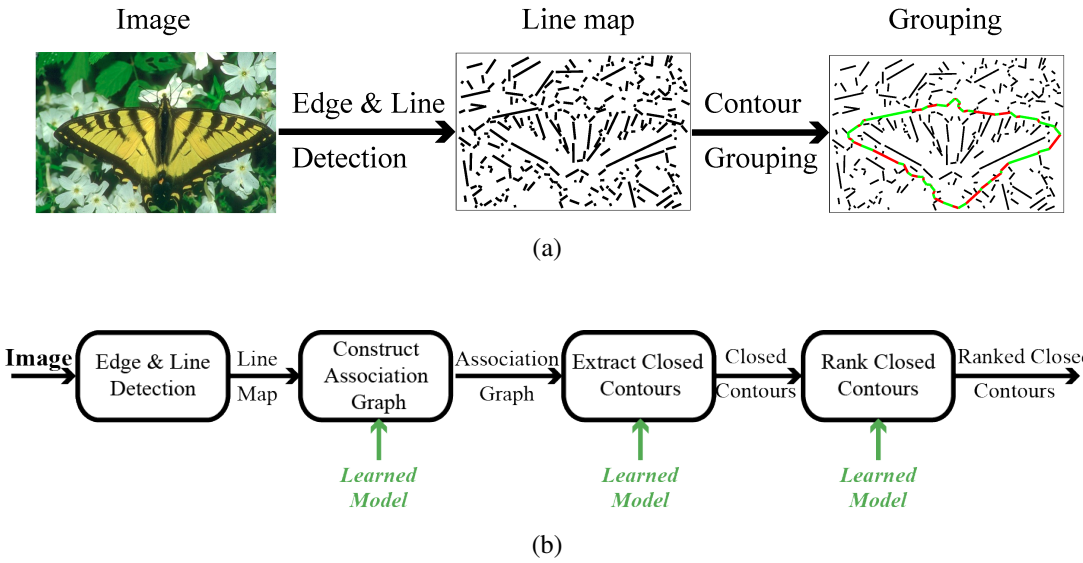



**Figure 1.2: Overview of algorithm**. (a) The output consists of multiple non-intersecting closed polygon comprised of visible line segments (green) connected by linear interpolants (red); (b) a schematic of the processing pipeline in our proposed method.

well a closed contour resembles the salient object in the image. This tool is used for two purposes: i) evaluating our method and comparing against other methods, and ii) providing a measure for evaluating contours in order to train models used in our method. Based on training samples, we learn to predict this error value for contour hypotheses constructed by our grouping algorithm.

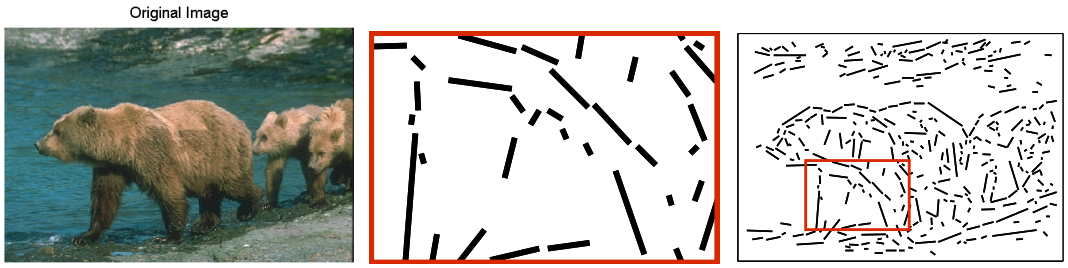
In addition to a performance measure, empirical evaluation of salient object segmentation methods also requires a dataset of ground truth object segmentations. ***Our first major contribution***, is to provide both a *ground truth dataset* and a *performance measure*. We have constructed a segmentation dataset called the Salient Object Dataset (SOD). The SOD is built upon the Berkeley Segmentation Dataset (BSD)[3], and provides ground truth boundaries of salient objects perceived by humans in natural images. We also psychophysically evaluated 5 distinct performance measures that have been used in the literature. Our results suggest that a measure based upon minimal contour mappings is most sensitive

to shape irregularities and most consistent with human judgements. In fact, the contour mapping measure is as predictive of human judgements as human subjects are of each other. Region-based methods, and contour methods such as Hausdorff distances that do not respect the ordering of points on shape boundaries are significantly less consistent with human judgements. We also show that minimal contour mappings can be used for Precision-Recall analysis. Our findings can provide guidance in evaluating the results of segmentation algorithms in the future.

2. Gestalt cues such as proximity and good continuation are often called 'local' cues, as they are defined on pairs of local oriented features. A Markov assumption (either explicit or implicit) is made in order to infer the probability or plausibility of longer chains of these features. While this Markov approximation has some statistical justification [5], it is an incomplete model [8], and more global cues must also be brought to bear in order to achieve good results [9, 10, 11].

   As an example, see Figure 1.3(b). It is hard to see how the line segments are related and how to group them on the boundary of the salient object in the image, without having a global view as in Figure 1.3(c). Accurate integration of these local and global cues is a challenge. ***A second major contribution*** of this dissertation is a novel, effective method for *combining local and global cues*, both at the stage of forming new closed contour hypotheses, and at the stage of evaluating and ranking these hypotheses.

3. Another challenge faced by contour grouping algorithms stems from the exponential nature of the search space, which leads to many very similar high probability hypotheses, while neglecting other more diverse hypotheses with slightly lower probability. This tendency lowers performance as partial hypotheses that look slightly less promising are weeded out too early. ***A third major contribution*** in this dissertation is a novel method for promoting *diversity* in the forma-




(a) (b) (c)

**Figure 1.3: Example of local and global information**. (a) A sample image (source: [3]), (b) local information available for grouping line segments, and (c) global information available for grouping line segments. It is easier to group line segments on the boundary of the salient object given the whole picture.

tion and ranking of contour hypotheses, leading to substantial improvements in performance.

4. To further improve the performance, a *multiscale implementation* of this method has been studied which obtains a spatial prior by running the contour grouping algorithm at a coarse resolution of the image and then uses this spatial prior in guiding the grouping search at a finer resolution. ***A fourth contribution*** in this dissertation is studying the effect of the multiscale prior on the performance and analysing the method for combining the results obtained in different resolutions.

5. The local cues used by our method have been used by others in prior work, but typically have been assumed conditionally independent and have been modelled parametrically as factored univariate distributions. ***Our final contribution*** is comparing the performance of these models with the use of a *multivariate mixture model* for the joint distribution. We obtain slight improvement by the mixture models.

## 1.4 Thesis Overview

After reviewing existing contour grouping literature in Chapter 2, I will elaborate on the above contributions in the following chapters:

- Chapter 3 introduces a new performance measure and dataset for evaluation of contours and contour grouping algorithms. In this chapter:
  - A new *Salient Object Dataset (SOD)* is introduced, built upon the widely used Berkeley Segmentation Dataset [3].
  - A new performance measure, called the *Contour Mapping Measure*, is introduced that respects the ordering of points on two shapes being compared.
  - We psychophysically evaluate 5 distinct performance measures that have been used in the literature.
  - We show our suggested measure is most sensitive to shape irregularities and most consistent with human judgements.
- Chapter 4 discusses the first stage of contour grouping, which is forming the Association Graph. In this chapter:
  - Preprocessing steps of edge detection and line approximation are discussed and compared.
  - Local Gestalt cues are modelled as both independent and dependent cues.
  - In order to capture statistical dependencies between local cues, the use of a *multivariate mixture model* for the joint distribution of cues is compared with univariate models, leading to very slight improvements in performance.
- Chapter 5 explains the second stage of contour grouping, which constructs closed contour hypotheses. In this chapter:
  - Local and global cues are selected for this constructive stage.

- These local and global cues are effectively combined in a novel *cost function* guiding the search.
- A novel method for maintaining the *diversity* of paths forming closed contour hypotheses is introduced.

- Chapter 6 introduces the last stage of contour grouping, which is ranking the closed contour hypotheses and selecting a few among them for output. In this chapter:

  - Effective cues are selected for this ranking stage.
  - These cues are combined in a new *saliency function* used for ranking the contours.
  - The effect of maintaining *diversity* in the output of the method is studied.

- Chapter 7 discusses the use of multiscale priors in the contour grouping method. These spatial priors are obtained from running the grouping algorithm at a coarse resolution. They are then used to guide the grouping algorithm at a fine resolution image. In this chapter:

  - Spatial prior cues are combined together with local and global cues to guide the constructive stage.
  - Spatial prior cues are combined together with ranking cues to guide the ranking stage.
  - The results of grouping in coarse and fine scales are combined effectively to yield the best results for each image.

In Chapter 8, we will compare our grouping method with existing methods. The last chapter will include discussions and conclusions.

# 2

# Related Research

Many researchers have tackled the problem of salient object segmentation. Algorithms developed during the 90's mainly worked in limited applications and under specific conditions (e.g. [12, 13], for example having simple dark backgrounds. Through the years, the methods have evolved into more advanced algorithms designed to operate in natural settings; yet the performance of currently available methods is far from ideal. The main difficulties are:

1. **Feature selection**. Working in the pixel space of digital images is not efficient due to the high-dimensional data space. Often higher level measurements or cues are used, for example gap along the object boundary, area enclosed by the object, maximum distance between pairs of points on the boundary of the object, number of convex parts, etc. However, it is not obvious exactly which cues should be used.
2. **Cue combination**. Even after a subset of available cues are selected in the image, it is not clear how they should be combined, For example is it better to combine the gaps along the boundary linearly or quadratically? or how is gap combined with color homogeneity?
3. **Finding the global optimum in a high dimensional search space**. Given an

ideal objective function designed by some combination of cues, an object can be extracted by finding the global optimum of this function. However, current objective functions lead to non-convex multidimensional search spaces, with lots of local optima, complicating the optimization process.

4. **Clutter and noise**. Natural scenes are usually cluttered, making design of robust algorithms hard.
5. **Occlusions**. Occluded or overlapping objects make the problem even more difficult.

The existing grouping methods vary in the levels of knowledge they utilize. Some approaches rely on some prior assumptions about the objects in the image, e.g. their type, shape, color, hue etc (e.g. [13, 14]). Some have access to additional information (e.g. depth) from motion, stereo, or video sequences (e.g. [15]). Some are interactive and get prior information from the user (e.g. [16]). Here, we focus on methods which do not use any prior knowledge or assumption about the objects or backgrounds, and work on a single still image without any user interactions.

Existing methods can be classified as either i) *local* or ii) *global*. In local approaches, the relationship strength between neighbouring elements does not depend upon the global context. In global methods, on the contrary, the relationship of two elements depends on the strength of relationships among other elements.

Some methods [17] use heuristics in solving the grouping problem, while some use probabilistic methods to learn models suitable for contour grouping given training data [18]. Depending on the way the problem is formalized, various optimization methods (e.g. shortest path methods, graph partitioning, etc.) have been used.

The outputs of grouping algorithms also vary. Some algorithms partition contour elements into unordered sets belonging to different objects existing in the image. Some also provide the ordering or sequence of these contour elements necessary to infer a curve. The closure constraint is also considered by some methods. Figure 2.1 shows a

brief overview of the contour grouping literature and its evolution during recent years.

In the following sections, I will review in more detail major methods that yield closed contours and are therefore more related to our contour grouping method. Among these, there will be examples of heuristic and probabilistic contour grouping methods.

## 2.1 Heuristic Methods

This section covers local grouping algorithms which model the grouping problem based on intuition and heuristics. Grouping costs or saliency measures are designed in a way to reflect intuition. For example, by the Gestalt factor of proximity [7], we know that we prefer to group tokens that are close together. Therefore, as an example, a saliency function can intuitively be defined to decrease when the sum of gaps between segments in a group increases. The saliency measure is used to compare a set of hypotheses and therefore guide the algorithm to the optimal grouping. Among heuristic methods, there are two major contour grouping methods: the Ratio Contour (RC) algorithm and the Adaptive Grouping (AG) algorithm.

### 2.1.1 Regional Ratio Contour Algorithm (RC)

The Regional Ratio Contour grouping algorithm of Stahl & Wang [19] is a simple grouping method that serves as a foundation for many subsequent studies (e.g. [11, 20, 21, 22]). This algorithm groups line segments detected in an image into a closed boundary corresponding to a a salient object in the image.

The boundary extraction problem can be modeled as an undirected graph $G = (V, E)$ where $V$ is the set of vertices and $E$ is the set of edges. For example, each endpoint of a detected fragment (which might be a line or a curve segment) is modeled as a vertex, resulting in an even number of vertices. Figure 2.2(a) shows a set of detected curve segments in an image. Virtual segments are needed between the detected

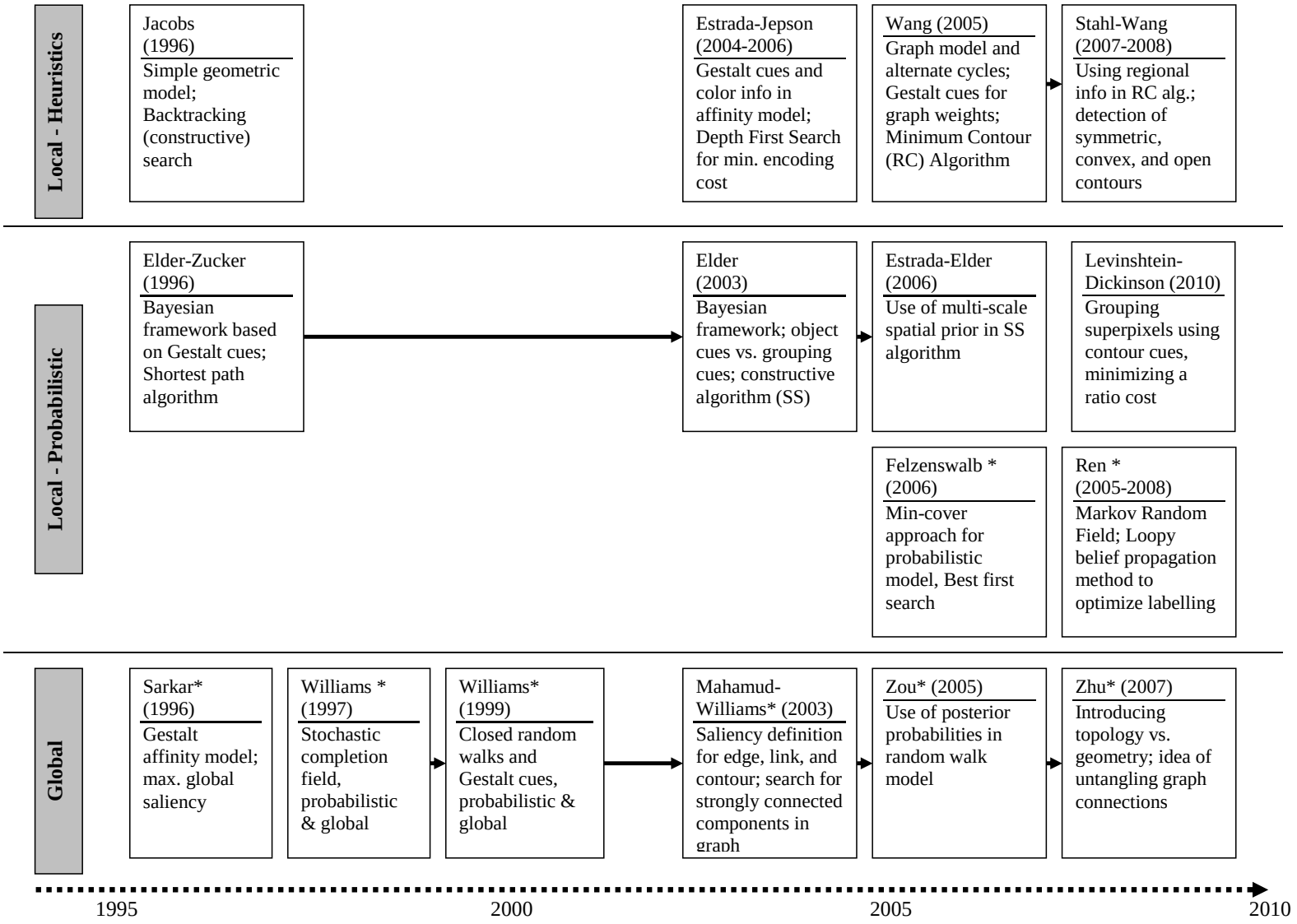



**Figure 2.1: Overview of recent major publications about contour grouping**. Some methods do not provide closure or sequencing (shown by *).

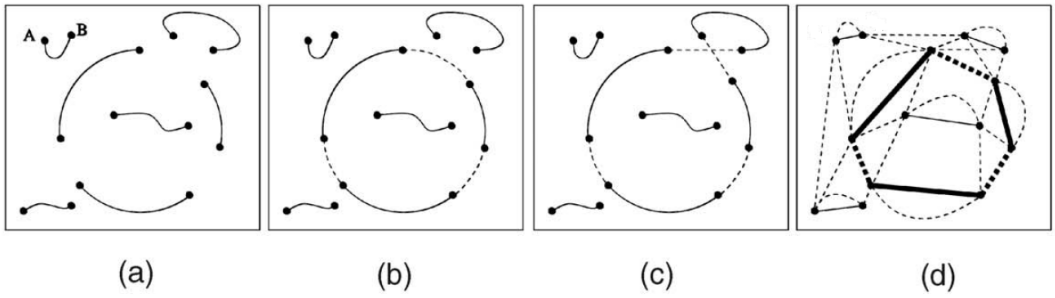



**Figure 2.2: Graph Model in Ratio Contour Algorithm.** (a) Real fragments corresponding to edges detected in an image. (b) Virtual fragments introduced between real fragments, and a non-degenerate closed contour passing through real and virtual fragments alternately, (c) A case of self- intersecting non-degenerate closed contour, (d) graph model of (a) and the alternate cycle corresponding to the contour in (b) shown in thick lines (reproduced from [17]).

segments, since not all points on the boundary of the object in the image are detected by the edge detector. An object boundary consists of a closed sequence of alternating detected edge segments and the virtual segments between them (Figure 2.2(b)). Moreover, object boundaries are non-degenerate (not passing through a segment more than once) and simple (not self-intersecting- see Figure 2.2(c)). In the model graph in Figure 2.2(d), real fragments are modeled as solid edges while virtual fragments are modeled as dashed edges. In this graph, a closed boundary is modeled as a cycle alternating between solid and dashed edges, and is referred to as an alternate cycle.

Enumeration of all possible cycles in this graph is not practical. Grouping costs are assigned as weights in the graph to lead the grouping algorithms towards detection of the optimal cycle. For example, grouping of fragments that are closer to each other is preferred to grouping those that are relatively far. Based on this type of heuristics, Jacobs [23] proposed a simple saliency measure defined as the ratio of the sum of lengths of detected fragments to the perimeter of the grouping cycle and found a set of salient closed contours using a greedy search algorithm.

Stahl and Wang [19] defined the ratio contour grouping cost for a boundary corre-

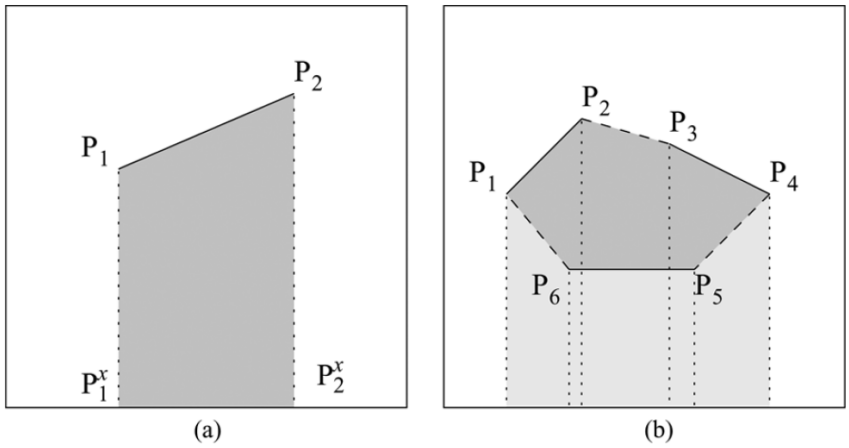



**Figure 2.3: Calculation of the enclosed area in the Regional Ratio Contour Method**- (a) Signed area defined for a line segment in the image. Note that when traversing this line from $P_1$ to $P_2$, the sign of this area is positive; and when traversing from $P_2$ to $P_1$, the sign of this area is negative. (b) The sum of the signed areas assigned to the line segments along the closed boundary equals the area of the region enclosed. (reproduced from [19])

sponding to a cycle $C$ in the grouping graph as:

$$\Gamma(C) = \frac{\sum_{e \in C} w_1(e)}{\sum_{e \in C} w_2(e)} \tag{2.1}$$

where each edge $e$ in the graph is assigned two weights: $w_1(e)$ and $w_2(e)$. The first weight is set to zero for solid edges. For dashed edges, this weight represents the Euclidean distance between the two endpoints of the corresponding virtual segment in the image and is a proximity cue. [1]

The second weight is defined as the signed area between segment $e$ and the $x$ axis, where the origin is defined at the left lower corner of the image. Therefore the absolute value of the sum of these weights for an alternate cycle in the graph equals the area of the enclosed region in the image, as shown in Figure 2.3. Note that for each segment, depending on the direction of traversal, the second weight can be positive or

[1] Although a smoothness cue was also added to the numerator in an earlier version of the RC method [17], they later [19] argue that incorporating a smoothness cue limits the applicability of the grouping method since many salient structures in real applications are not always smooth.

negative. For detected (solid) line segments in the image, the signed area is divided equally between the two adjacent dashed edges of the graph, resulting in zero weight for these edges. For a closed simple boundary in the image, the absolute value of the cost $\Gamma$ (ignoring the direction of traversal) translates to the ratio of total gap along the boundary divided by the area of the region enclosed inside the boundary.

Finding the minimum cycle in a doubly weighted graph is a theoretical problem studied by Lawler [24] and Karp [25]. In the Ratio Contour algorithm, this problem has been reduced to finding a negative weight cycle in a graph as follows:

Consider the following transformation of the grouping graph $G$ to produce the reweighted graph $G'$, with:

$$w'_1(e) = w_1(e) - bw_2(e), \forall e \in E \tag{2.2}$$

where $e$ is an edge on the graph in the set of edges $E$. Note that the above transformation will not change the ranking of cycles, i.e. for two cycles $C_1$ and $C_2$ where $\Gamma(C_1) < \Gamma(C_2)$ , we still have $\Gamma'(C_1) < \Gamma'(C_2)$ where the cycle ratio $\Gamma'$ is calculated using the transformed weights. If cycle $C^*$ has the minimum cycle ratio among all cycles in graph $G$, it will have the minimum transformed cycle ratio among all cycles in graph $G'$, in other words $\forall C : \Gamma(C^*) < \Gamma(C) \rightarrow \forall C : \Gamma'(C^*) < \Gamma'(C)$. For a certain parameter $b^*$ the transformed cycle ratio for $C^*$ is zero: $\Gamma'(C^*) = 0$, while all other cycles will have positive transformed cycle ratios. Note that $b^* = \Gamma(C^*)$.

Assuming that we have a detector for negative weight alternate (NWA) cycles in a graph (as will be explained shortly), the parameter $b^*$ can be found by a sequential search. First $b$ is set to a high value so that all cycles in the graph will have negative total weights. Then it is gradually decreased until no cycle with a negative total weight can be detected in the graph, as shown in the following algorithm:

**Ratio Contour Sequential Search Algorithm**

1. Initialize $b = \max_{e \in E} \frac{w_1(e)}{w_2(e)} + 1$. This is an over estimate for $b*$.
2. Transform the edge weights to obtain graph $G'$.
3. Then use the Negative Weight Alternate (NWA) cycle detection algorithm to detect an NWA cycle as follows:
   (a) Find a minimum weight perfect matching (MWPM) $P$ in $G'$.
   (b) Obtain one negative cycle $C$ from P.
4. If C is a NWA (i.e. $\Gamma'(C) < 0$ ), calculate the cycle ratio $\Gamma(C)$ using the original edge weights (i.e. without applying the edge-weight transformation). Set $b$ to $\Gamma(C)$ and go to Step 2.
5. If no NWA cycle is detected, return the alternate cycle $C$ detected in the previous iteration as the minimum ratio alternate cycle $C*$ in $G$.

The negative weight alternate cycle (NWA) detector in Ratio Contour algorithm uses minimum weight perfect matching (MWPM). A perfect matching in graph $G' = (V, E')$ is a subgraph $P = (V, E')$ containing all the vertices, but only a subset of edges $E' \subset E$ such that every vertex is paired with exactly one other vertex by an incident edge. A minimum weight perfect matching (MWPM) is a perfect matching with the minimum total edge weight among all possible perfect matchings. In the graph $G'$ constructed above, the set of all solid edges denotes a perfect matching with a total weight of zero. The MWPM $P$ will have a total weight less than this trivial perfect matching and therefore has a nonpositive total weight. Polynomial time algorithms exist for finding the minimum weight perfect matching, e.g. [26], and are used in step 3(a) in above algorithm.

The next step is to obtain the cycle $C$ corresponding to this MWPM. This can be done by removing all solid edges and their adjacent vertices from $P$, and then adding all solid edges that were not originally in $P$ to obtain a new graph $P'$ (see Figure 2.4). It has been proven that if the total weight in the MWPM is negative, then $P'$ has at least one negative cycle [17].

The time complexity of Ratio-Contour (RC) algorithm is shown to be $\mathrm{O}\big(|V|^{3/4}|E|\big)$.

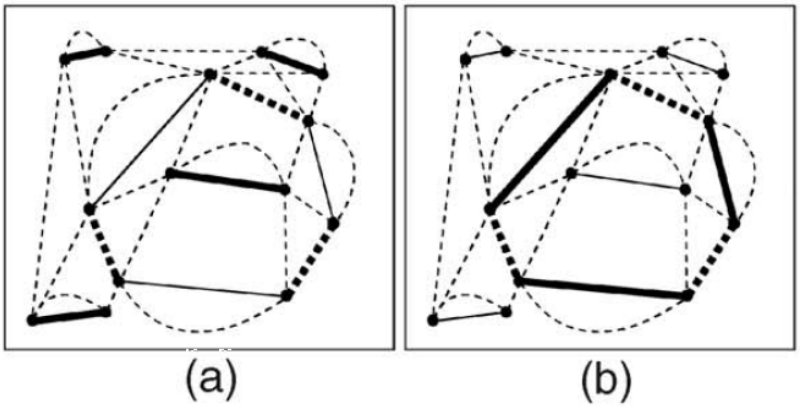



**Figure 2.4: Finding the negative weight alternate (NWA) cycle in the Ratio Contour Algorithm**- Given a minimum weight perfect matching (MWPM) in the graph, shown by thick lines in (a), a negative weight alternate (NWA) cycle can be obtained, shown by thick lines in (b), as explained in the text. (reproduced from [17])

If virtual segment construction is limited to a factor of the number of vertices, i.e. $|E| = \mathrm{O}\big(|V|\big)$ , the overall complexity will be $\mathrm{O}\big(|V|^{7/4}\big)$ [17]. The polynomial time complexity of the RC algorithm is one of the most important features of this algorithm.

Figure 2.5 shows some sample results. Note that although the region area term in the denominator of the grouping cost usually leads the algorithm towards simple boundaries, the RC algorithm is not guaranteed to return simple contours. Self-intersections are not common among the results of this algorithm. In fact, Stahl et al. [19] reported only one non-simple boundary among 4500 experiments.

Stahl and Wang have also suggested many other variations of the above algorithm using symmetry [21], convexity [11], and occluded boundaries [22].

### 2.1.2 Adaptive Grouping Algorithm (AG)

Another simple geometric approach to the grouping problem with competing performance has been outlined by Estrada and Jepson [27]. This method is also based on heuristics, and has been shown to have a better performance than an earlier version of the RC method. This method uses proximity, smoothness, and color similarity in its

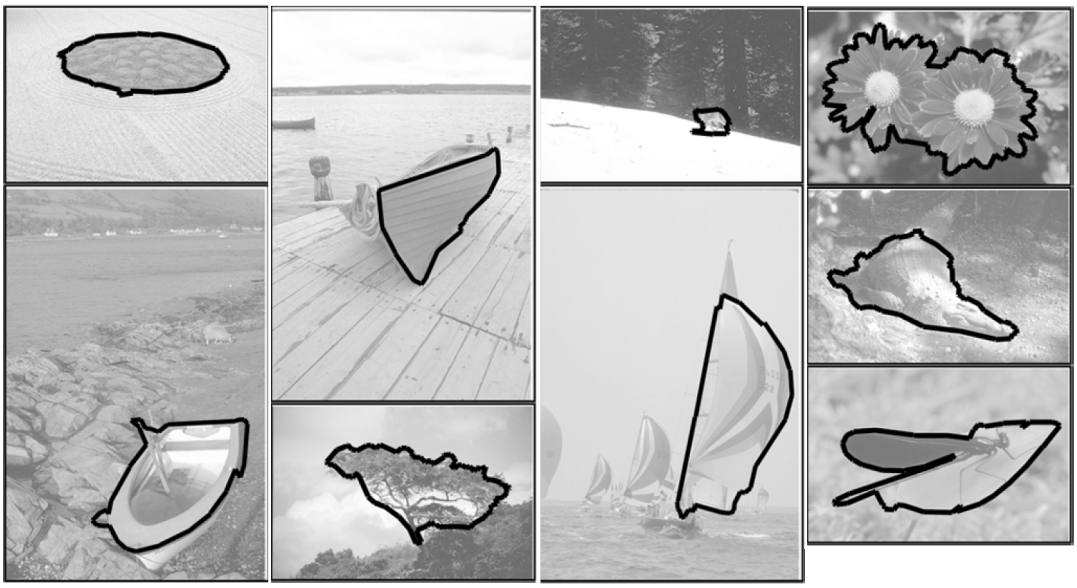


**Figure 2.5: Sample results of the Regional Ratio Contour Method** (reproduced from [19])

grouping cost function.

Consider a sequence of fragments $s$ ending with fragment $i$. The affinity $T$ of a fragment $j$ with $s$ is defined as:

$$T(s,j) = G(i,j)C(s,j) + \gamma \tag{2.3}$$

where $G(i,j)$ is a geometric measure based on proximity and good continuation between fragments $i$ and $j$, designed heuristically to promote sequences with smaller gaps and smooth continuation; and $C(s,j)$ is a measure of color histogram similarity between two sides of $s$ and $j$. Both $G$ and $C$ are designed to have values in the interval $[0,1]$ and $\gamma$ is a suitable positive constant (e.g. $\gamma = 0.1$ ). Based on the affinity values calculated as above, a subset $\kappa$ of best k tangents are chosen as candidates for extending $s$. The normalized affinity is calculated as:

$$N(s,j) = \frac{T(s,j)}{\sum_{k \in \kappa} T(s,k)} \tag{2.4}$$

The affinity values are only used to obtain acceptable grouping hypotheses, and are not directly used to define the saliency of a contour.

In the Adaptive Grouping algorithm, sequences are generated in a greedy depth-first search. Only extensions resulting in affinity values higher than a threshold are considered. Unlike the Ratio Contour cost, the affinity function of AG does not promote compact boundaries, therefore in the greedy search, sequences with compactness lower than a threshold are discarded, where compactness is defined as the area of the region enclosed inside the boundary divided by the area of its convex hull. It is not clear how the "inside" region is defined for open sequences[1]. In every iteration of the search algorithm, self intersecting sequences are discarded, resulting in only simple closed contours. Colour histograms are pre-computed to speed up the algorithm.

After all the closed contours are generated, they are ranked based on their saliency. The saliency of a closed sequence $s$ is calculated as the sum of negative log probabilities of drawing a pixel $x$ in the image from inside and outside of the region $S$ bound by the contour sequence $s$:

$$\text{Saliency}(s) = -\sum_{x \in S} log(p(x|H_{\text{in}}) - \sum_{x \notin S} log(p(x|H_{\text{out}}) \tag{2.5}$$

where $H_{\text{in}}$ and $H_{\text{out}}$ represent color histograms of the whole inside and whole outside regions respectively. This measure assigns a lower cost to boundaries that have homogeneous inside and outside regions, and are therefore less expected to wander into textured and cluttered regions. Figure 2.6 shows a few sample results. The AG algorithm can be summarized as follows:

[1]This is not mentioned in their paper [27], and only the executable version of their code is publicly available.

**Figure 2.6: Sample results of the Adaptive Grouping Method** (reproduced from [27])

**The Color Adaptive Grouping Algorithm**

Initialize $C = \{\}$.

1. Compute color histograms $H$ at each pixel in image using windows of fixed size $w_{\text{size}}$ and a fixed number $n_{\text{bins}}$ of bins.
2. For each fragment $i$,
   (a) Generate an ordered sequence containing only fragment $i$ , i.e. $s = (i)$.
   (b) Find candidate set $\kappa$ for extending s.
   (c) For each fragment $j \in \kappa$
      i. Calculate normalized affinity $N(s, j)$.
      ii. If $N(s, j) < \tau_{affty}$ go to next $j$,
      otherwise add $j$ to the end of $s$, setting $s' = (s, j)$.
      iii. If $s'$ has any self-intersections, discard $s'$ and go to next $j$.
      iv. If Compactness$(s') < \tau_{compact}$ , discard $s'$ and go to next $j$.
      v. If $s'$ is a closed sequence, add $s'$ to set $C$ and go to next $j$.
      vi. Recursively go to step (b) with $s = s'$.
3. Calculate saliency values for all cycles in $C$, and return the best.

## 2.2 Probabilistic Methods

Contour grouping can be perceived as a problem of probabilistic inference. In a probabilistic method, a probability value should be assigned to each hypothesized contour to provide a measure of confidence in it. A contour around a salient object in an image should have a high probability, while contours not corresponding to object boundaries

should have lower probability values. The properties of the segments, e.g. the contrast between their sides, as well as relationships between segments, e.g. their proximity, good continuation and similarity can be used as cues. Each cue provides evidence towards a reliable probabilistic inference.

There is a long history of research into the relationship between human perception and the statistics of the visual world. Attneave (1954) [28] and Barlow (1961) [29] proposed that the role of the early visual system is to reduce statistical redundancy therefore increasing coding efficiency. Brunswik and Kamiya (1953) [30] suggested the Gestalt principles of perceptual grouping should be tuned to the statistics of the natural world. This was confirmed by experiments done by Kruger [31], Sigman et al. [32] and Geisler et al. [33]. Martin et al. [3] also studied the use of natural image statistics for image segmentation.

Elder and Goldberg [5] used object contours traced by human observers in natural images to estimate the statistics of Gestalt cues such as proximity, good continuation, and similarity. These statistics were used by Elder and Krupnik [9] to guide a search for highly probable closed contours within a Bayesian framework. This approach has been extended to a multi-scale framework using a coarse-to-fine search strategy with improved results [10]. Other related probabilistic approaches are Felzenszwalb's min-cover [34] and the Markov Random Field approach by Ren et al. [8, 35]. Unlike the algorithms proposed by Elder and colleagues [9, 10], the latter approaches do not group contour fragments into simple closed contours, and are therefore not discussed further here.

### 2.2.1 A Probabilistic Framework for Contour Grouping

Given N tangents $t_k, k = 1..N$ in an image and assuming $C$ as the set of all contours in the image, let $C^o \subset C$ denote the set of all *object contours*. A contour $c$ is a cycle of contour elements tangent to an underlying curve. A candidate contour of length $n$

is a cycle of the following form:

$$c = \left(t_{\alpha_1}, t_{\alpha_2}, ..., t_{\alpha_{n-1}}, t_{\alpha_n}\right) \in C^{o*} \subset C^o,$$
$$\alpha_i \in 1..N, i \in 1..n, \alpha_i \neq \alpha_j \text{ if } i \neq j \text{ except } \alpha_1 = \alpha_n \tag{2.6}$$

The sequence $c$ is an n-tuple and $C^{o*}$ is the set of all simple (non self-intersecting) closed object contours possible in the image.

A probabilistic grouping algorithm searches among the closed contour candidates for the sequence $c^*$ whose probability is maximum given the cues D.

$$c^* = \arg\max_c p(c \in C|D) \tag{2.7}$$

The set of cues $D$ provides grouping evidence based on observed measures, for example the proximity of tangents, their similarity, etc. In [9, 10], two types of cues are suggested:

1. Unary (or object) cues ($D^o$): These cues provide evidence about whether a tangent should be included in the contour. For example, if an object is known to have certain color, the similarity of the colour on either side of a tangent with the expected colour can be used as a cue. The grouping likelihood assigned to the $i^{\text{th}}$ tangent, $t_i$, given the $k^{\text{th}}$ unary cue is denoted by $p(d_i^k|t_i \in T^o), d_i^k \in D^o$, where $T^o$ is the set of tangents.

2. Binary (or grouping) cues ($D^c$): These cues provide evidence about the ordering of tangents in the contour sequence. They show how probable it is for one tangent to be positioned right after another tangent. In other words they show the strength of relationships between the two tangents. The Gestalt cues of proximity, good continuation, and similarity are all examples of this type of cue. The likelihood of tangent $t_j$ being grouped with tangent $t_i$ given the $k^{\text{th}}$ binary cue is

denoted by $p(d_{ij}^k|\{t_i,t_j\} \in C), d_{ij}^k \in D^c$.

The following assumptions simplify the computation of probabilities of contours given above cues:

- **Markov Chain assumption:** By this assumption the likelihood of a pair of tangents grouping depends only on these two tangents and the relations between them, and is independent of all other tangents in the image. This assumption is supported by the fact that the strongest statistics lie in the relations between directly successive tangents on the contours and decrease substantially for distant tangents [5]. By limiting the Markov neighbourhood to a pair in the sequence, grouping will be pair-wise independent. Note that this assumption is only an approximation [8].

- **Independence of evidence provided by different cues**: Based on the observation that the correlation among cues is low, the evidence provided by the cues can be assumed to be independent [5].

Based on the above simplifying assumptions and also assuming that we are comparing contours of the same length [9], the posterior probability of a contour $c$ can be written as:

$$p(c \in C^o|D) \propto \prod_{t_i \in T^o} p_i^o \prod_{\{t_i,t_j\} \in C} p_{ij}^c \tag{2.8}$$

where

$$p_i^o = \frac{1}{1+(L_i^o P_i^o)^{-1}}, \quad p_{ij}^c = \frac{1}{1+\left(L_{ij}^c P_{ij}^c\right)^{-1}} \tag{2.9}$$

$$L_i^o = \prod_{k=1}^{l_o} \frac{p\left(d_i^k|t_i \in T^o\right)}{p\left(d_i^k|t_i \notin T^o\right)}, \quad P_i^o = \frac{p\left(t_i \in T^o\right)}{p\left(t_i \notin T^o\right)} \tag{2.10}$$

$$L_{ij}^c = \prod_{k=1}^{l_c} \frac{p\left(d_{ij}^k|\{t_i,t_j\} \in C\right)}{p\left(d_{ij}^k|\{t_i,t_j\} \notin C\right)}, \quad P_{ij}^c = \frac{p\left(\{t_i,t_j\} \in C\right)}{p\left(\{t_i,t_j\} \notin C\right)} \tag{2.11}$$

Parameter $l_o$ is the number of unary (or object) cues and $l_c$ represents the number of binary cues. The prior ratios $P_i^o$ and $P_{ij}^c$ and likelihood ratios $L_i^o$ and $L_{ij}^c$ can be learnt from the statistics of training images and ground truth contours. Note that this method does not have any free parameters and all the required parameters are learnt from training data and image statistics.

For example in [9], the goal is to detect boundaries of lakes in satellite images. Using boundaries hand drawn by mapping experts for some training images, the probability distributions for tangents on the lake boundary and off the lake boundary can be learnt for cues such as the intensity on the dark side of the tangent. If an object model is available, cues such as distance of tangent to the nearest point on the model and the angle of tangent w.r.t. the nearest model segment can also be used. Another unary cue used in the above probabilistic framework by [10] is the boundary energy (or Pb, probability of boundary) introduced by Martin [36] which represents brightness, colour, and texture contrasts.

#### 2.2.1.1 Graph Model

The above problem can be modelled as a directed graph $G = (V, E)$, similar to that used by Stahl and Wang, where $V$ is the set of vertices and E is the set of edges. Each tangent $t_i$ is modeled as a vertex $v_i$, and each link or virtual segment between $t_i$ and $t_j$ is modeled as a directed edge $e_{ij}$ between the two vertices $v_i$ and $v_j$.[1] Each edge $e_{ij}, i = 1...N, j = 1..N$ in graph $G$ is assigned a weight $w^c(e_{ij})$ which is indicative of the (binary) probability of grouping the adjacent vertices (or equivalently, probability of grouping the corresponding tangents). Moreover, each vertex $v_i, i = 1..N$ is assigned a weight $w^o(v_i)$, which is indicative of the (unary) probability of including the

[1] This simplifies the graph model used in RC where each tangent was modeled by two vertices for the two endpoints and thus two types of links, solid and dashed, were defined to differentiate between line segments and their links.

corresponding tangent in the grouping. The weights are calculated as follows:

$$w^c(e_{ij}) = -\log(p^c_{ij}), i, j = 1..N, \quad w^o(v_i) = -\log(p^o_i), i = 1..N \tag{2.12}$$

A path $P = (v_{\alpha_1}, ..., v_{\alpha_m})$ in the graph from $v_{\alpha_1}$ to $v_{\alpha_m}$ is equivalent to an ordered sequence of tangents $s = (t_{\alpha_1}, ..., t_{\alpha_m})$ and its corresponding weight (or grouping cost) is calculated by:

$$w(P) = -\left(\sum_{v_i \in P} w^o(v_i) + \sum_{\{v_i, v_j\} \in P} w^c(e_{ij})\right) \tag{2.13}$$

Comparing the above Equation with 2.8, we can see that this weight is proportional to the negative log probability of the contour.

To reduce the complexity of graph construction and graph search, only the best neighbours of each tangent are considered, and therefore each vertex $v$ will have $k$ outgoing edges[1], similar to RC and AG methods. The complexity of constructing this sparse graph is therefore reduced from $\mathrm{O}\big(N^2\big)$ to $\mathrm{O}\big(N\big)$ for N detected segments in the image.

#### 2.2.1.2 Searching for Closed Contours in the Graph

Since the weight of a path is proportional to its negative log probability, the maximum probability sequence has a minimum sum of weights. Therefore the problem is reduced to finding the shortest cycle in the above graph. Dijkstra's shortest path algorithm has been used by Elder and Zucker [37] to solve this problem. However, Elder and Krupnik [9] observed that this method does not in general yield a simple contour, and is biased to computing smaller contours. They propose an alternative search algorithm that chooses tangents to be included in a sequence in a greedy way. This method does

[1] Only $k = 10$ extensions are considered in [9]

not guarantee an optimal solution, but can handle the above issues. The following constructive algorithm is suggested in [9]:

**The Constructive Probabilistic Algorithm**

Initialize $C = \{\}$; $m = 1$; $S = \{s_1 = (t_1), s_2 = (t_2), ..., s_N = (t_N)\}$ as set of all paths of length 1 (all nodes).

1. Loop
   (a) Extend each path $s_i$ in $S$ by one node. Set $m = m + 1$.
   (b) Discard all paths corresponding to non-simple (self-intersecting) curves.
   (c) Add all closed paths to set $C$, and remove them from $S$.
   (d) Calculate the cost of each path in $S$: $W(S) = \{w(s_1), ..., w(s_{n_m})\}$.
   (e) Sort by W and select best $N_m = \frac{N_{mem}}{m}$ sequences from $S$. Discard the rest.

   Until $m = M$ ( a maximum length for paths, predetermined based on learned distribution of object boundary lengths)
2. Calculate posterior probabilities for all cycles in $C$, and return the best.

In each iteration, the sequences explored in the previous iteration are extended by one tangent. All virtual links from the last tangent, or equivalently all outgoing edges from the corresponding node are considered to convert the sequence $s_i$ of length $m$ to a new sequence of length $m + 1$. The new sequences are then checked for self-intersection and closure. Those with self-intersection are discarded. Closed sequences are set aside as closed contour hypotheses. When a maximum number of iterations is reached, these closed contours will be ranked and the best are selected for output.

In addition to limiting the virtual links during the construction of the graph, only $N_m = \frac{N_{mem}}{m}$ sequences are kept in each iteration.[1]

[1] This value is set to $N_{mem} = 4000$ in [9].

#### 2.2.1.3 Contour Saliency

In the last step of the above algorithm, the posterior probability of each cycle is calculated as a measure of saliency for the closed contours of varying length. The grouping probability of a cycle defining an object in the image is expected to be high. In fact three factors must be considered in the posterior to select a closed sequence $s$ as the best contour: 1) A foreground factor, $F^*$, measuring the probability that $s$ is a salient contour; 2) A background factor, $B^*$, which measures the probability that all other tangents are actually in the background and not on the boundary of the object; and 3) A prior factor, $P^*$, accounting for the prior on the number of tangents in $s$ given the number of tangents in the image. The background factor reduces the bias of the algorithm towards shorter contours. The posterior is calculated as

$$p(s = c^*|D) = \frac{p(D|s = c^*)p(s = c^*)}{p(D)} = F^*(s, D)B^*(s, D)P^*(s) \tag{2.14}$$

This grouping method has been successfully used to compute exact lake boundaries from high resolution satellite imagery [9]. This framework also has the capability to use available knowledge about the shape, size, color, or location of objects. For example skin color information was used to segment skin regions in [9]. Obviously the algorithm is less successful when operating without any prior knowledge about the object in the image. The simplifying assumptions used in this framework such as independence of cues, Markov assumptions, and assuming there is only one salient object in the image and everything else is background, are only approximations of the true probabilistic model of grouping. These assumptions can result in poor performance in realistic situations. Moreover, this algorithm does not exploit any global information.

### 2.2.2 The Multiscale Grouping Algorithm (MS)

The multi-scale algorithm of Estrada and Elder [10] uses a pyramid of scaled images, propagating contours from coarser resolutions of the image to finer resolutions. At the coarsest scale, the single scale algorithm outlined in the previous section, is applied. The contour obtained at this scale is upsampled, using Fourier Descriptors, to a finer scale where it is used as a spatial prior. This is repeated through scale space until a contour is derived at the finest scale (see Figure 2.7).

Several issues are addressed by the multi-scale approach:

- Complexity: The search space is smaller at the coarse scale, and thus the grouping algorithm is more likely to find good approximate solutions.
- Global Constraints: Many grouping algorithms, such as the single scale probabilistic method outlined in the previous section, rely on the Markov Chain assumption for its simple structure. Yet depending only on local information results in missing the big picture, or "missing the forest for the trees" [10]. Using the spatial information at a coarse scale to guide grouping at a fine scale captures this global information.
- Noise and clutter: Smaller, less important details are eliminated at coarse scales, reducing distractions.

To exploit the coarse scale closed contours as spatial priors, two spatial cues have been suggested [10]: 1) The distance between a tangent and the prior, and 2) the angle between the tangent and the prior. As before, the likelihood distributions for these spatial cues are learnt from ground truth data for tangents on and off the object boundaries.

The multi-scale algorithm was found to enhance the performance of the probabilistic grouping algorithm[10]. See Figure 2.8 for sample results.

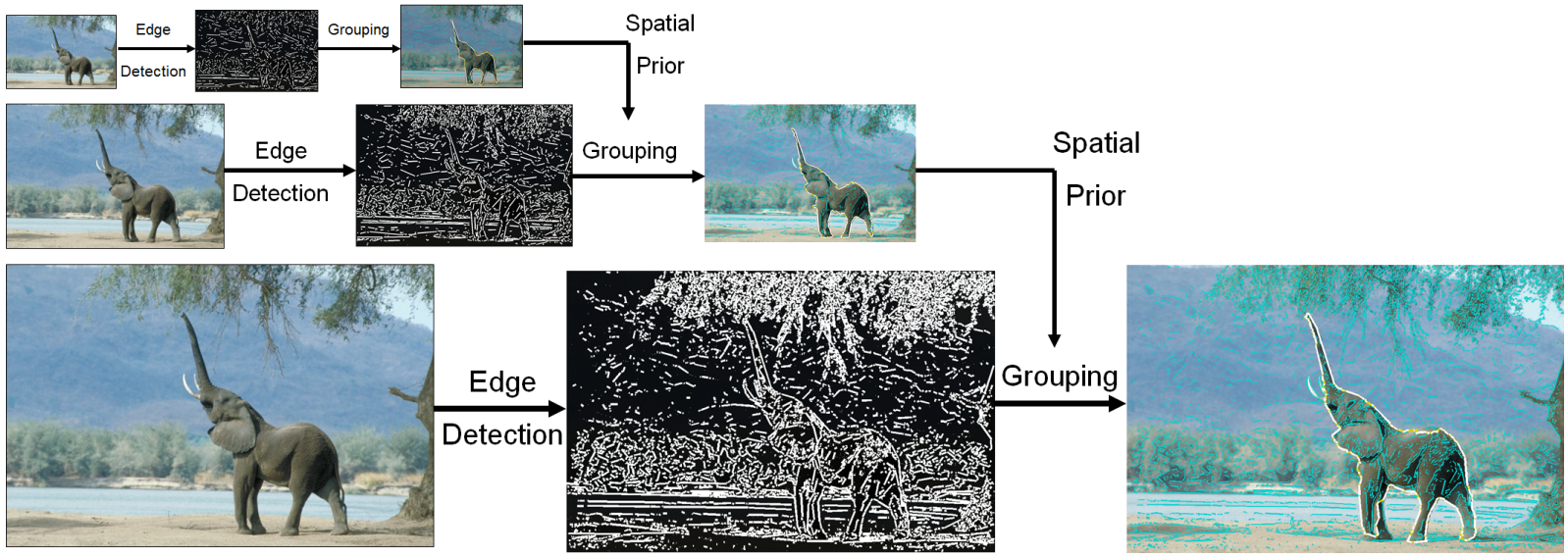



**Figure 2.7: Overview of the Multiscale contour grouping algorithm.** Spatial priors are obtained from coarse scales and used as spatial cues at finer scales (reproduced from [38]).

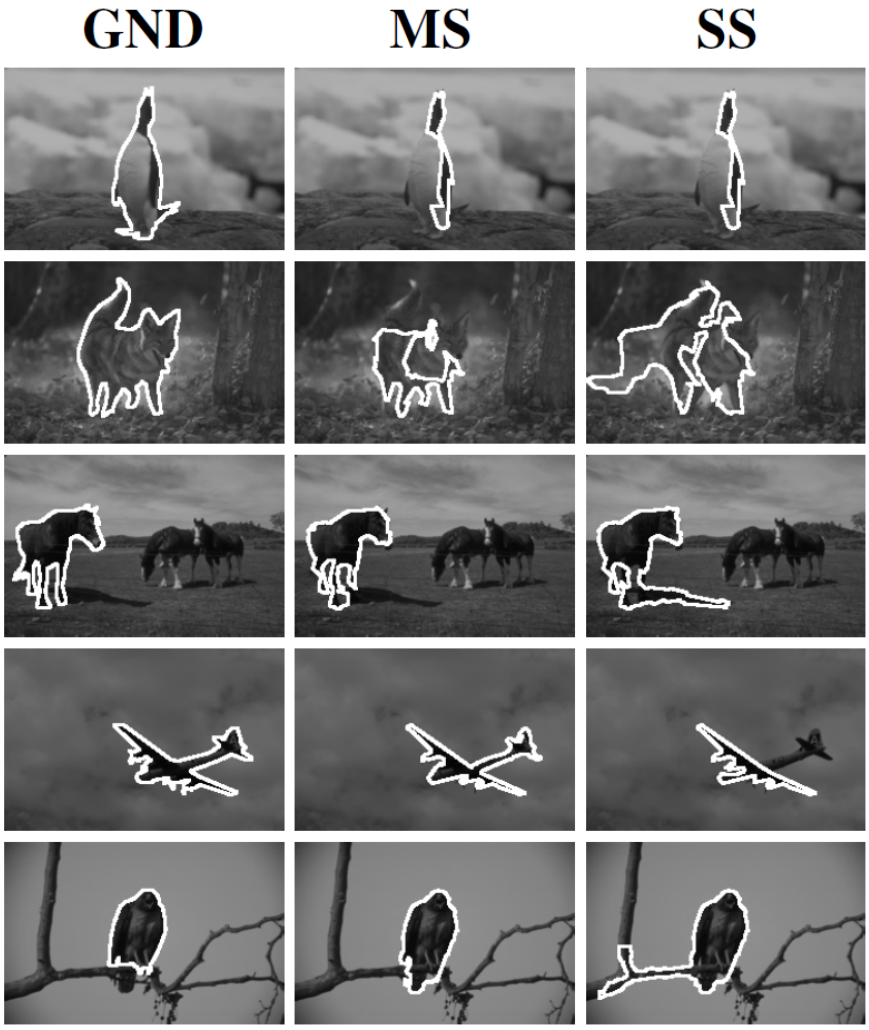



**Figure 2.8: Comparing the Multi-scale and Single-scale probabilistic methods.** The first column (GND) shows the ground truth boundaries for a few images from the BSD [3] database. The 2nd column (MS) shows the resulting contours using the multi-scale probabilistic algorithm, while the 3rd column (SS) shows the results of the single scale probabilistic algorithm (reproduced from [10]).

### 2.2.3 The Superpixel Closure Algorithm (SC)

In the Superpixel Closure algorithm of Levinshtein et al. [20] the problem of finding closed contours as cycles of edge fragments is reformulated as the problem of finding spatially coherent subsets of superpixels which have strong image edge evidence. By grouping superpixels instead of edge fragments, the grouping method benefits from lower search complexity. Moreover, the search is faster since there is no need to check for self-intersections.

Assuming a preprocessing step of superpixel segmentation on the image, $X_i$ is defined as a binary indicator variable for the $i$-th superpixel. The value of this variable can indicate whether the superpixel is selected as figure (1) or ground (0). A vector $\mathbf{X}$ can therefore define a full labeling of all superpixels in the image. Similar to the ratio cost of Stahl and Wang [19], the closure cost is defined as $C(\mathbf{X}) = \frac{G(\mathbf{X})}{A(\mathbf{X})}$, where $G(\mathbf{X})$ indicates the total gap along the outer boundary of "on" superpixels, and $A(\mathbf{X})$ denotes the area defined by them.

The above cost function is decomposed into unary and binary terms as follows:

$$C(\mathbf{X}) = \frac{\sum_i G_i X_i - 2\sum_{i<j} G_{ij} X_i X_j}{\sum_i A_i X_i} \tag{2.15}$$

where $A_i$ is the area of the $i$-th superpixel, $G_i$ is the total gap along the $i$-th superpixel's boundary, and $G_{ij}$ is the gap along the shared boundary between the $i$-th superpixel and the $j$-th superpixel (see Figure 2.9).

Each of the gap components are further defined as $G_{ij} = P_{ij} - E_{ij}$. If $EP_{ij}$ is the set of pixels on the shared boundary of the two superpixels, then $P_{ij} = |EP_{ij}|$ is the number of pixels on the shared boundary. $E_{ij}$ is a measure indicating the edginess of these pixels and is defined as $E_{ij} = \sum_{p \in EP_{ij}} E_{ij}^p$ where $E_{ij}^p = [L(f^p) > T_e]$ is an edge indicator for pixel $p$, assigning a 0 or 1 to the pixel based on a logistic regressor $L(.)$ trained on a set of features $f^p$ and given a threshold $T_e$. The set of features for each

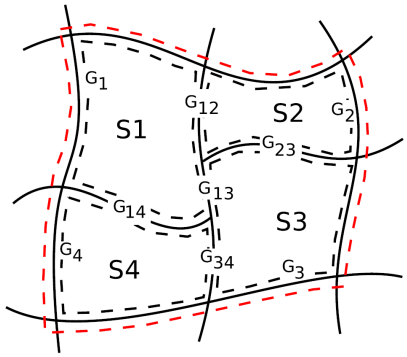



**Figure 2.9: Grouping cost of the Superpixel Closure Algorithm.** The gap term in the grouping cost is decomposed into unary and binary terms. The unary gap terms $G_i$ corresponds to the total gap along the boundary of superpixel $S_i$, while the binary term $G_{ij}$ is the boundary gap along the shared edge between superpixel $S_i$ and superpixel $S_j$. The total gap along the boundary of the selection (shown in red) is calculated as $G_{1234} = \sum_{i=1}^{4} G_i - 2(G_{12} + G_{13} + G_{14} + G_{23} + G_{34})$ (reproduced from [20]).

pixel on the super pixel boundary includes:

1. Distance to the nearest edge in the image
2. Strength of the nearest edge
3. Alignment between edge orientation and tangent to superpixel boundary at the pixel
4. Squared curvature of the superpixel boundary at the pixel

Using 30 hand labelled images from the Weizmann Horse Database [39], the above logistic regressor was trained using the above features. The threshold value $T_e$ determines the number of groups being detected. Decreasing its value results in more grouping hypotheses to be generated, and facilitates the detection of small objects. However, low values are reported to hurt the performance of the method if the number of allowed output groupings is limited.

The above cost function can be optimized using an optimization method such as the optimal graph cut based method of Kolmogorov et al. [40] which can have a polynomial run time for finding the global optimum.

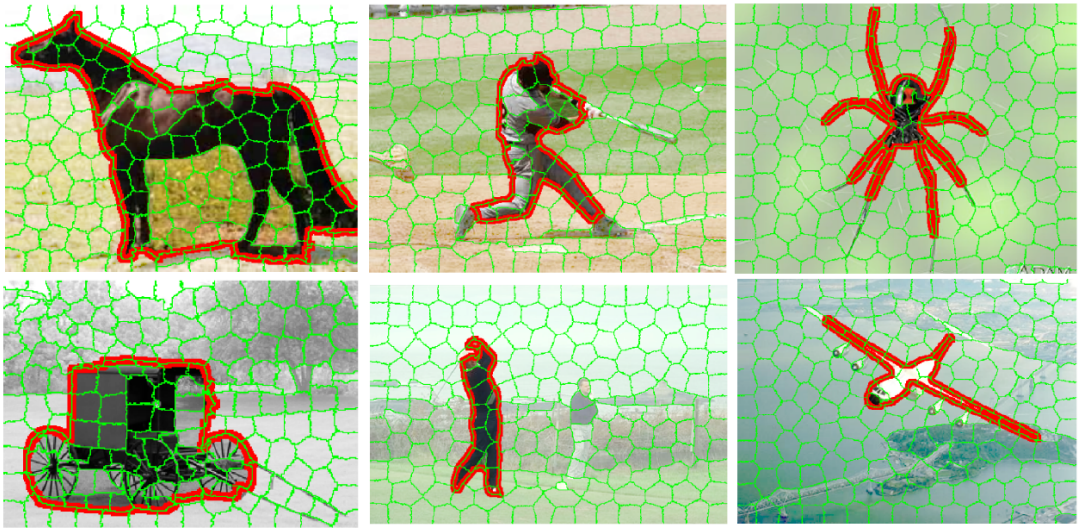


**Figure 2.10: Sample results of the Superpixel Closure Method** (reproduced from [20]).

The authors compared their method with the grouping methods of Stahl and Wang [19] (RC) and Estrada and Jepson [27] (AG) and showed better or same performance on images from Weizmann Segmentation Database [41] and Weizmann Horse Database [39] based on an F-measure defined by regional intersection of algorithm and ground truth boundaries. A few sample results are shown in Figure 2.10.

## 2.3 Global methods

In global methods, the weights or costs of possible groupings of segment pairs are not assigned locally and are therefore not independent of the weights assigned to other groupings or links. These global weights are sometimes called the saliency of the edges or links. The *saliency* of an edge/link is defined as a value which is correlated with whether that edge/link belongs to a shape or is part of background noise [42]. The cost or saliency of the contour is then calculated using these values, which are now globally defined.

Global methods are mainly based on *stochastic completion fields* and *closed ran-*

*dom walks* introduced by Williams [42]. The saliency functions defined by random walks were used by Mahamud et al. [43] to construct a graph model in which the strongly connected components represent groupings. Zou et al. [44] and Zhong et al. [45] used the most salient edge as the grouping seed or starting point in their grouping methods. Zhu et al. [46] studied the problem of "entanglement", i.e. the existence of strongly connected edge pairs in an image that are not in agreement with the optimal grouping. They used a graph model and a circular embedding in the complex domain to identify and detect such edges in order to isolate simple closed curves.

Although the above methods show a good performance in finding salient edges, these salient edges have not been grouped into salient object boundaries. In other words, these implementations are more focused on measuring saliency and less on salient object segmentation. Similarly, recent methods [47, 48] that find the "objectness" of sliding windows in the image are designed to highlight regions in the image that have a higher probability of belonging to an object. However, they also do not perform salient object segmentation.

## 2.4 Conclusion

Various models have been used to formulate the grouping problem. Although some are based on heuristics and intuition, others are based on more rigorous probabilistic frameworks. Due to the complexity of the problem and the multimodal search space, simplifying assumptions and approximation methods have been used which result in suboptimal groupings. Another reasons for suboptimal performance of available algorithms is failure to exploit global information. Suggested global methods are not designed for salient object segmentation. There is the need for efficiently combining local and global cues in a probabilistic framework.

# 3

# Data Set and Evaluation

As mentioned in the previous section, the goal is to perform *object segmentation*, i.e. to extract the *simple closed contour* bounding the *salient object* in the image. An important question is how to measure success. How can we evaluate performance of object segmentation methods?

The history of evaluating segmentation algorithms is as old as the history of segmentation algorithms themselves. Zhang [49] published a survey on these methods in 1996. He has classified the evaluation methods into three main categories:

1. *The analytical methods.* These evaluation methods consider the algorithms without considering their output. The major difficulty in evaluation by analytical methods is the lack of a general theory for image segmentation.
2. *The empirical goodness methods.* These evaluation methods are based on the outputs of the segmentation algorithms. For example the outputs can be compared based on the intra-region uniformity of the segments, or the inter-region contrast between the segments. Although these evaluation methods may be suitable for segmentation of images into uniform regions, they are not suitable for evaluation of object segmentation (or figure-ground segmentation) approaches, since there is no reported goodness measure that can generalize to segmentation

of all images. See for example Figure 3.1(a) where the intra-region uniformity and the inter-region contrast are both low.

3. *The empirical discrepancy methods.* In discrepancy evaluation methods, a reference segmentation or ground truth is assumed. The outputs of the segmentation algorithms are then compared with the ground truth. Due to the limitations of previous evaluation methods, we will focus on this type of evaluation, i.e. evaluation by discrepancy or error measures.

In order to use empirical discrepancy methods to evaluate performance of segmentation algorithms, there is the need for i) a dataset of ground truth boundaries of salient objects, and ii) a suitable error measure for the task of salient object segmentation. These will be discussed in the following sections. In section 3.1, we will discuss the need for a ground truth dataset and will introduce the Salient Object Dataset (SOD). Then in section 3.2, we will review the previously suggested error measures and will introduce our new measure. These error measures are compared based on psychophysical experiments, reported in section 3.3.

The contents of this chapter have been published in the 7th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision (POCV10) [50].

## 3.1 The Salient Object Dataset (SOD)

The most common method for obtaining ground truth data is using human judgements. Yet even using this method, there are major differences found in the segmentations provided by different human subjects. Humans produce segmentations at different granularities and with different levels of detail, even when they perceive the image as having the same hierarchical structure [3].

For the purpose of object segmentation, existing ground truth datasets used previously for performance evaluation have limitations. For example, the Berkeley Seg-

mentation Dataset (BSD) [3] provides a suitable set of 300 images and their ground truth segmentation, segmented by up to 30 subjects. Yet these segmentations do not generally correspond to the boundaries of salient objects in these images. The segments usually correspond to areas in the image with homogenous color or texture, and not necessarily to regions corresponding to objects in the image. Moreover there is no distinction between foreground and background segments. (see for example Figure 3.1(a)). The PETS dataset [51], Goldmann's dataset [52], and the ground truth dataset of lake boundaries used in [9] are limited in their domain of images.

Ge's ground truth dataset [1] contains figure-ground segmentation of 1023 natural images from the internet, digital photos, and image databases with the most salient foreground structure segmented. Two subjects worked together to extract one object boundary. Another similar dataset, known as the Weizmann Segmentation Dataset [41], contains 200 images with objects that differ from their surroundings by either intensity, texture, or other low level cues. Since limited to having only one or two salient objects, these images are manually segmented into two or three segments by 3 human subjects. A pixel was declared as foreground if it was marked as foreground by two of the subjects. These datasets are limited in two ways: 1) Since objects were deliberately selected to have high contrast with the background, the dataset may be biased. 2) The dataset does not contain information about the variation in perceived segmentation over subjects.

In order to overcome these limitations while taking advantage of prior work, our new SOD dataset is constructed based on human segmentations in the BSD300 [3]. A set of human subjects were employed to identify the salient object in 300 natural images in BSD. Variations over both BSD subjects and SOD subjects for the same image provides a reasonable measure of subject variability.

Figure 3.1 shows examples of the visual interface. The Salient Object Dataset (SOD) was constructed from the judgements of 7 human subjects (graduate students).

(a) (b)

**Figure 3.1: Example objects from the SOD dataset** - (a) A sample segmentation from BSD where several segments compose one salient object in SOD, (b) Another sample in SOD with more than one salient object.

Each subject was presented with a random subset of the BSD boundaries superimposed on the corresponding image. All 300 images of the BSD were employed, with about half of the BSD segmentations shown to each subject. The subject was asked to identify the object(s) he or she perceives as most salient by clicking on the BSD segment(s) that comprised each object. The subject could combine several segments to form one object (3.1(a)), and also could identify multiple salient objects in the same image (3.1(b)). In the latter case, subjects were required to rank the identified objects according to their salience. This resulted in a total of 12,110 salient object boundaries selected by 7 subjects.

The SOD dataset is publicly available at `http://www.elderlab.yorku.ca/SOD/` and has been used by other researchers for evaluation of salient object segmentation [53, 54, 55].

## 3.2 Error Measures

The second requirement for performance evaluation is having a suitable error measure. The error measures used in the literature can be categorized into i) region-based mea-

sures, ii) boundary-based measures, iii) mixed measures. We will review a number of previous proposals and identify potential weaknesses. Based on this analysis, we propose a new measure. The method finds the optimal mapping between the two point cycles forming the boundaries of the 2D shapes being compared, where optimality is defined in terms of the sum of Euclidean distances between corresponding points. We show that it is important that the correspondence is monotonic, respecting the ordering of the points. We demonstrate how this method addresses issues arising with previous methods.

### 3.2.1 Region-based Error measures

Region-based error measures consider the consistency between the regions (or pixels) comprising algorithm and ground truth segments (Figure 3.2). For example, the *regional coincidence accuracy* proposed by Ge et al. [1] is defined as an "Intersection over Union" measure:

$$\text{IoU}(A;B) = \frac{|R_A \cap R_B|}{|R_A \cup R_B|} \tag{3.1}$$

where $R_A$ and $R_B$ are the pixels within algorithm segment $A$ and ground truth segment $B$ respectively, and $|.|$ returns the number of pixels. A *region intersection (RI)* measure is then given as

$$RI(A,B) = 1 - \text{IoU}(A;B) \tag{3.2}$$

Region-based measures are usually symmetric with respect to the two segments and therefore treat false positives and false negatives in the same way. Other examples of region-based measures counting the number of misclassified pixels include the Negative Rate Metric [51, 52], the Hamming Distance [56], the Local Consistency Error [3], the Bidirectional Consistency Error [57], and the regional Precision-Recall measures [3].

The regional measures are not optimal for evaluating segmentation algorithms as

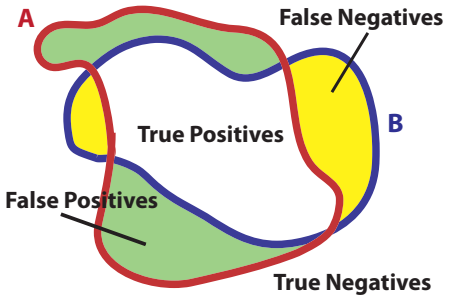
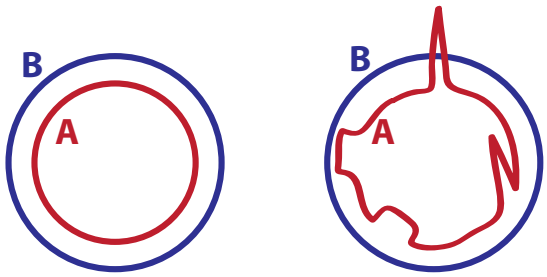



**Figure 3.2: Region-based error** between algorithm ($A$) and ground truth ($B$) segments.




(a) (b)

**Figure 3.3: Limitations of region-based measures**- Regional measures are not sensitive to spikes, wiggles and some large shape features. Based on regional measures, the boundary in (a) is almost as good as the boundary in (b) when compared with the ground truth circle, although it has spikes, wiggles and shape differences.

they are not sensitive to spikes, wiggles, and major shape differences. For example, most region-based measures would predict that the algorithm segments (A) in Figure 3.3 (a) and (b) are comparable in their consistency with ground truth (B), whereas to the human eye, the algorithm result is better in (a) than in (b).

### 3.2.2 Boundary-Based Error Measures

Boundary-based measures evaluate segmentations based on the accuracy of their boundaries. The algorithm boundary is compared with a ground-truth boundary. The error measure is usually some aggregate measure of distance between points on the two boundaries. In particular, for each point $a$ on the boundary $A$, a distance to boundary

$B$, denoted as $d_B(a)$, can be defined as the minimum distance of point $a$ to all points on $B$.

$$d_B(a) = \min_{b \in B} \left( d(a,b) \right), \qquad a \in A \tag{3.3}$$

where $d(a,b)$ is a distance measure between the points $a$ and $b$.

Consideration of the distance of all points in A from B yields a *distance distribution signature* ($SD$) [57]: $SD_B(A,B) = \{d_B(a), a \in A\}$. In a similar fashion, $d_A(b), b \in B$ and $SD_A(B,A)$ can be defined. Aggregate values of these distributions can be used as a measure of distance. For example, letting $D_B(A,B)$ represent the mean distance of points on the boundary of A from B, i.e. $D_B(A,B) = \overline{SD_B(A,B)}$, leads to a *mean distance (MD)* error measure:

$$MD(A,B) = \frac{1}{2} \left( D_B(A,B) + D_A(B,A) \right) \tag{3.4}$$

When a measure with less sensitivity to outliers is needed, the median of the distribution can also be used.

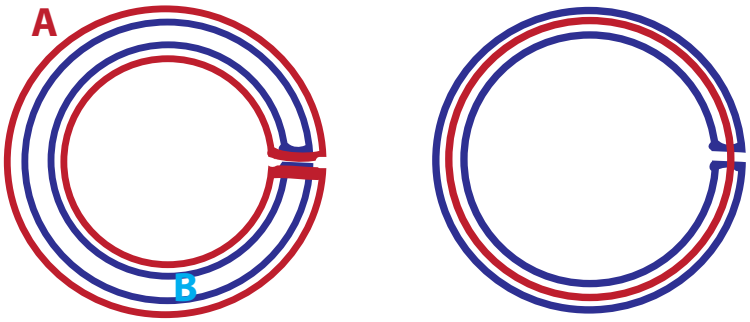
The *Hausdorff distance (HD)* [58], on the other hand, is based upon the maximum value of the distance distribution signatures :

$$h(A,B) = \max \left( SD_B(A,B) \right) = \max_{a \in A} \min_{b \in B} d(a,b) \tag{3.5}$$

$$HD(A,B) = \max \left( h(A,B), h(B,A) \right) \tag{3.6}$$

Since the Hausdorff distance only looks at the maximum value in the distance distribution, two contours having the same worst case distance are evaluated as being the same, irrespective of other distances.

While these boundary-based measures do not suffer from the problem depicted in Figure 3.3, they are subject to a different problem. In Figure 3.4, these boundary-based measures would assign similar errors to the algorithm boundaries ($A$), whereas the algorithm boundary in (b) has far greater perceptual error.




(a) (b)

**Figure 3.4: Problems with boundary-based distance measures** - Ground truth ($B$) is shown in blue and algorithm contours ($A$) are shown in red. The segmentation in (a) is reasonable, but the segmentation in (b) is grossly incorrect. However, standard boundary measures would assign similar error to (a) and (b).

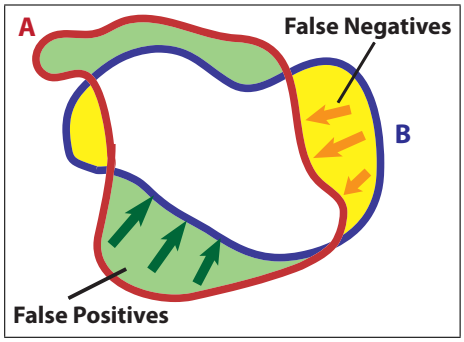



**Figure 3.5: Measures using a mixture of regional and boundary information** - The false positive and false negative regions are penalized by their distance from the intersection region.

### 3.2.3 Mixed Measures

One can attempt to solve the shortcomings of region and boundary based error measures by combining the two to form a *mixed measure (MM)* as follows (see Figure 3.5):

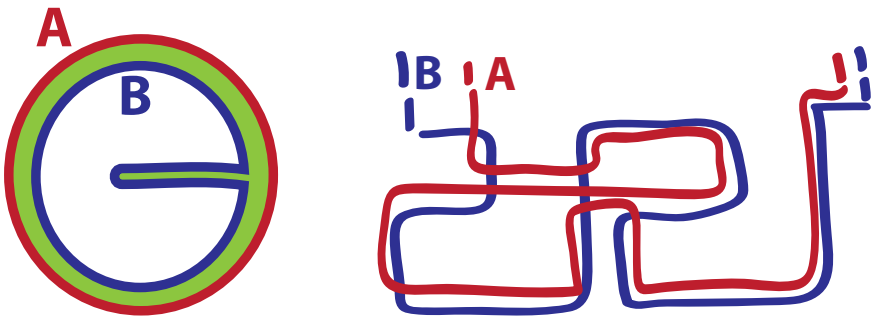
$$MM(A,B) = \frac{1}{2D_{diag}} \times \left( \frac{1}{N_{fn}} \sum_{j=1}^{N_{fn}} d_A(p_j) + \frac{1}{N_{fp}} \sum_{k=1}^{N_{fp}} d_B(q_k) \right) \tag{3.7}$$

where $N_{fn}$ is the number of false negative pixels, and $d_A(p_j)$ is the distance of the jth false negative pixel, $p_j$, from the algorithm boundary $A$. Similarly $N_{fp}$ is the number of false positive pixels and $d_B(q_k)$ is the distance of the kth false positive pixel, $q_k$, from the ground truth boundary $B$. $D_{diag}$ is the diagonal size of the image and can be used to normalize the distance values. Other error measures suggested in this category are the rate of misclassification metric [51] and the weighted quality measure metric [51].

Although the above measures are sensitive to wiggles and spikes, they still are not sensitive to some important shape differences. For example, in Figure 3.6(a) the green region is penalized by distances to the intersection area. Since the pixels in this region are close to B, the above measures do not effectively penalize the difference in the shapes. A more exaggerated case is shown in Figure 3.6(b). These examples demonstrate that very bad segmentations can be assigned very small error measures. The core problem appears to be that these measures do not enforce a direct monotonic mapping of boundary points, allowing the measure to remain low even when the shapes diverge.

### 3.2.4 Contour Mapping Measure

Elastic matching methods[59, 60, 61, 62, 63, 64] directly align two contours by determining a mapping between the points on the contours that minimizes a matching cost. Typically, the matching cost is based on two components: 1) dissimilarity of local properties of matched points, e.g. tangent orientations, and 2) dissimilarity of matched




(a) (b)

**Figure 3.6: Mapping limitations** - The mixture measures cannot penalize all shape differences effectively.

curve segments, i.e., the cost of deforming one curve segment (stretching, bending or compressing) to match the other curve segment. Equipped with a translation-, rotation- and scale-invariant cost function, these measures have proven effective in image database search applications and for clustering of shape databases [65]. Optimization is often based on cyclic string correction [66] methods and its variants [62, 65]. The contour mapping measure we propose is in the spirit of these elastic measures. The cost function is simply the Euclidean distance of matched points.

Following the notation of Maes et al. [66], we represent shape boundaries $A$ and $B$ as strings of points, $A = a_1 a_2 ... a_n$, $B = b_1 b_2 ... b_m$. A mapping between point $a$ and point $b$ is denoted by $s : a \leftrightarrow b$ (Figure 3.7). To avoid the problems illustrated in Figures 3.4 and 3.6, the order of the mapping must be monotonic. In other words, if $a_i \leftrightarrow b_m$ and $a_j \leftrightarrow b_n$ then $i < j \Rightarrow m \leq n$ and $m < n \Rightarrow i \leq j$. For closed boundaries, the indices are assigned cyclically.

Note that although the mapping is monotonic, it is not necessarily strictly monotonic, and thus need not be 1:1. The boundaries being compared can have different levels of detail and very different total arc lengths; $1 : n$ and $n : 1$ mappings allow the relative speeds of the two curves to vary accordingly while maintaining correspondence. Tagare et al. [67] call this class of correspondence a bimorphism (Figure 3.7).

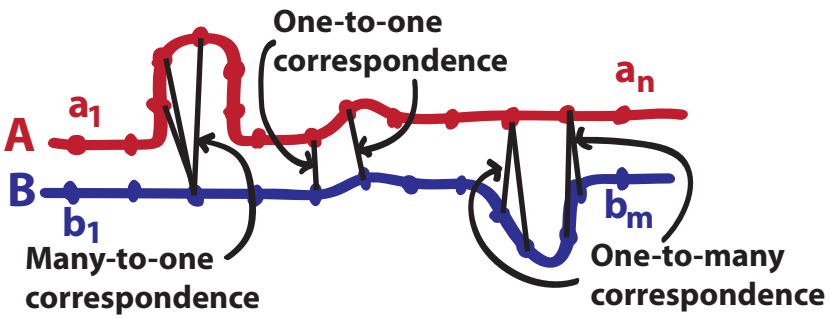



**Figure 3.7: Bimorphism**

We define a mapping sequence $S = s_1 s_2 ... s_k$ as a mapping between $A$ and $B$ in which all points in $A$ are mapped to at least one point in $B$ and vice versa. The cost of this sequence is $\gamma(S) = \sum_{i=1}^{k} \gamma(s_i)$, where $\gamma(s_i)$ is simply the Euclidean distance between the points. The mapping distance, $\delta(A, B)$ , is defined as the minimum cost of mapping $A$ and $B$ [66]:

$$\delta(A, B) := \min_{S} \gamma(S) \tag{3.8}$$

A trace $T$ from $A$ to $B$ is the set of $k$ ordered pairs of integers $(i, j), i \in 1..n, j \in 1..m$ corresponding to the $k$ mappings in a mapping sequence. Since all points on A and B have a match, we have:

$$\forall i \in [1..n], \exists j' \in [1..m] : (i, j') \in T \text{ and}$$

$$\forall j \in [1..m], \exists i' \in [1..n] : (i', j) \in T \tag{3.9}$$

Potential mappings and associated costs can be represented as a graph (Figure 3.8). Moving down the graph corresponds to advancing on the boundary $A$ and moving right on the graph is equivalent to advancing on the boundary $B$. The set of points traversed on a path from the upper left corner of this graph to the lower right corner defines a trace starting from $(a_1, b_1)$ and ending at $(a_n, b_m)$ and represents the mapped point pairs on the two boundaries. Such a path ensures that all points on the two boundaries have a match. The monotonicity condition constrains each edge of the path to have only down and/or rightward components. Since the total matching cost is defined as a sum, if edges are weighted by the matching costs the shortest path from $(a_1, b_1)$

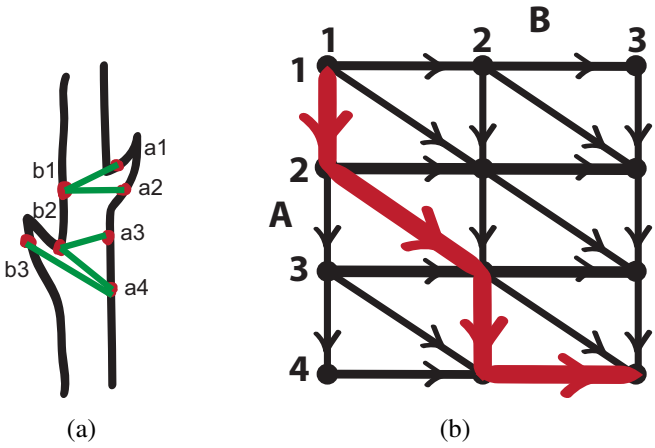



**Figure 3.8: Mapping graph for calculating CM** - (a) Two boundaries $A = a_1a_2a_3a_4$ and $B = b_1b_2b_3$. (b) Associated mapping graph. The red path corresponds to the sequence of mappings $S = (a_1 \leftrightarrow b_1, a_2 \leftrightarrow b_1, a_3 \leftrightarrow b_2, a_4 \leftrightarrow b_2, a_4 \leftrightarrow b_3)$ and the trace $T = (1,1),(2,1),(3,2),(4,2),(4,3)$ of size $|T| = 5$ with mapping distance of $\gamma(S) = d(a_1,b_1) + d(a_2,b_1) + d(a_3,b_2) + d(a_4,b_2) + d(a_4,b_3)$. If this sequence has the lowest mapping distance among all possible mapping sequences between $A$ and $B$, then $\delta(A,B) = \gamma(S)$.

to $(a_n, b_m)$ corresponds to the minimum cost matching. Since the mapping costs are symmetric, i.e. $\gamma(a \leftrightarrow b) = \gamma(b \leftrightarrow a)$, the mapping distance is also symmetric and we have $\delta(A,B) = \delta(B,A)$.

In the preceding discussion, we assumed that the first (and last) points on the two boundaries were matched. Since the points on the boundaries of the two shapes form cycles, we must consider all possible cyclical shifts of the boundaries. A cyclical shift $\sigma^k$ of size $k$ of the boundary $A = a_1a_2...a_n$ is defined by $\sigma^k(a_1a_2...a_n) = a_{k+1}...a_na_1...a_k, 1 \leq k \leq n$, and $\sigma^0(A) = A$. The equivalence class of $A$ defined by $k$ cyclic shifts will be denoted by $[A]$. Therefore:

$$\delta(A,[B]) := \min \delta\left(A, \sigma^l(B)\right), 0 \leq l < m \tag{3.10}$$

We define the *contour mapping measure (CM)* as the normalized mapping distance between the boundaries $A$ and $B$:

$$CM(A,B) = \frac{1}{|T|}\delta(A,[B]) \tag{3.11}$$

where $T$ is the trace corresponding to the optimal mapping sequence and $|T|$, the size

of the trace, is the number of mapped point pairs.

The distance $\delta(A, B)$ can be obtained by shortest path methods in the mapping graph, as explained above, or can be solved using dynamic programming, since the problem can be broken into sub-problems as follows. We define $A_i = a_1...a_i$ and $B_j = b_1...b_j$. We have $\delta(A_1, B_1) = \gamma(a_1 \leftrightarrow b_1) = d(a_1, b_1)$. For $i \in [2..n]$ and $j \in [2..m]$, we have:

$$\delta(A_i, B_j) = d(a_i, b_j) + \min \begin{cases} \delta(A_{i-1}, B_{j-1}) \\ \delta(A_{i-1}, B_j) \\ \delta(A_i, B_{j-1}) \end{cases} \tag{3.12}$$

Using Dynamic Programming to find $\delta(A, B)$ has a complexity of $O(mn)$ since the distance calculation between $n$ points on $A$ and $m$ points on $B$ is $O(mn)$ and the dynamic programming table itself is of size $mn$. Assuming $m \leq n$, constructing the same table for the $m$ cyclic shifts of $B$ will result in a complexity of $O(m^2 n)$. Using a method similar to the method proposed by Maes [66] for string editing, the complexity can be reduced to $O(nm \log m)$.

By requiring explicit monotonic correspondence between points on the two shapes, the contour mapping measure (CM) avoids the problems experienced by other boundary measures (Figures 3.4 and 3.6), in which the error of very different shapes is minimized by implicitly mapping the same or nearby points on one shape to points that are widely separated (in arc-length) on the other curve.

## 3.3 Psychophysical Experiments

Mumford [68] raised this question: *"There are many mathematical ways to define a numerical measure of the similarity of 2 shapes: do any of these approximate the human idea of similarity?"* Here we report the results of two psychophysical experiments that

address this question. Specifically, we compare human judgements of shape similarity with decisions made based on the region intersection measure (RI) (Eq. 3.2), mean distance (MD) (Eq. 3.4), Hausdorff distance (HD) (Eq. 3.6), mixed measure (MM) (Eq. 3.7), and the contour mapping measure (CM) (Eq. 3.11).

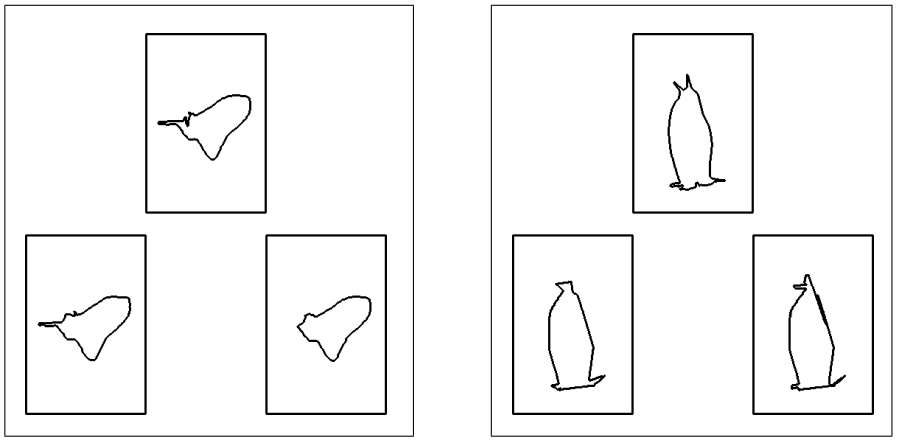
### 3.3.1 General Methods

Both experiments consisted of a set of trials in which the subject was shown a reference shape $A$ and two test shapes $B$ and $C$ (Figure 3.9), and asked to indicate which of the test shapes appeared more similar to the reference. The shapes were drawn from 30 of the 300 images in the BSD/SOD database that contain at least one completely unoccluded salient object. The shapes were displayed as outlines. (In preliminary experiments we found that judgements were similar for outlines and silhouettes).

In both experiments the reference shape was an object segmentation from SOD. The two experiments differed only in the nature of the test shapes. In Experiment 1, the test shapes were other segmentations of the same object by other human subjects. In Experiment 2, they were machine-generated approximations of the reference shape. We used these two very different sets of stimuli in order to judge how well the results are likely to generalize.

The 9 subjects who participated in the two experiments were naïve to the exact purpose of the experiments. There was no time limit: subjects could view the shapes for as long as they wanted.

In order to assess the consistency of each measure described in Section 3.2 with human judgements, we ran each measure as an 'observer' for the two experiments. On each trial, each of the measures was used to compute the similarity of the two test shapes to the reference shape, and the test shape with the higher similarity was 'selected' by the measure. Agreement with human subjects was then computed as the percentage agreement in the test shape selected over all trials.



(a) (b)

**Figure 3.9: Psychophysical displays** - In both experiments, the reference shape A was shown at the top of the display, and the two test shapes B and C were shown at the bottom. (a) Experiment 1: reference and test shapes are segmentations of the same object by different human observers. (b) Experiment 2: reference is a human segmentation, while test shapes are approximations generated by an automatic algorithm. See text for details.

### 3.3.2 Experiment 1- SOD hand drawn boundaries

*Stimuli*: The three shapes within each trial were selected from different human segmentations of the same object in 30 images of the BSD. The shape differences were thus due to inter-subject variations in the original BSD segmentations and/or the SOD constructions. All segmentations in each image were considered. Two segmentations were considered to be of the same object if they had at least 10 percent regional overlap (by the RIM measure). For each object $O$, the complete set $L_O$ of possible stimulus triplets (reference plus two test shapes) was created, based upon the segmentations for that object. This yielded a large set $L = \bigcup_O L_O$ of candidate stimulus triplets to use for the experiment.

We then took two steps in order to select from $L$ a subset of stimulus triplets that would maximize the discriminative power of the experiment. First, we selected the subset of triplets that generated disagreement between at least one pair among the 5 error measures considered, since inclusion of pairs on which all measures agree would

not serve to discriminate the measures.

We observed that some measures (e.g. MM) had more cases of disagreement with CM than others, and choosing stimuli randomly from L would not include enough samples of disagreements of other measures. To be able to compare measures fairly, we needed the same number of stimuli for each of 10 possible pairs of disagreeing measures. For a simpler implementation and to be able to compare CM fairly with other measures, we only looked at the 4 pairs where CM disagreed with another measure and required that the number of samples from each of these four (overlapping) subsets be (approximately) the same.

Since many segmentations in SOD are very similar and hard to distinguish by eye, our second step selected from each of the four subsets, the triplets for which test shapes $B$ and $C$ were maximally different. To achieve these goals, dissimilarity of test shapes $B$ and $C$ was measured by both $CM$ and the competing measure $M$ with which it disagreed. The two distance measures were converted to z scores and summed. The 50 triplets generating the highest z-score sum for each measure $M$ were then selected. The union of the resulting 200 triplets yielded the $170$ unique triplets that were ultimately used in the experiment.

*Results*: Figure 3.10(a) shows the overall consistency of each measure with the human subjects. The results show that the contour mapping measure (CM) is the most consistent with human judgements among all five measures. Since each of the 9 subjects saw the same stimuli, we could also compute an overall average consistency between our human subjects (pink dashed line). Remarkably, the CM algorithm is as predictive of human judgements as human subjects are of each other. The HD measure is the closest competition in this experiment. Figure 3.10(b) shows pairwise comparisons between the CM measure and each of the alternative measures on the subset of trials on which they disagreed. These differences are all statistically significant at the $\alpha < .05$ level (Table 3.1).

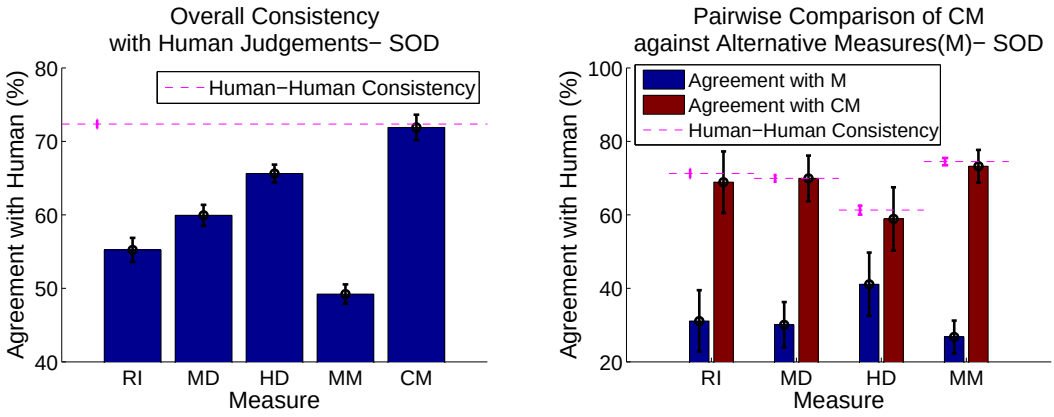



**Figure 3.10: Results of Experiment 1** - (a) Overall consistency with human subjects. (b) Consistency with human subjects for trials on which CM disagrees with each other measure $M \in \{RI, MD, HD, MM\}$. Error bars indicate standard error of the mean.

| Experiment | RI | MD | HD | MM |
|---|---|---|---|---|
| SOD | 1.4e-4 | 1.1e-5 | 1.5e-2 | 2.8e-7 |
| ALG | 5.7e-2 | 1.6e-1 | 2.8e-3 | 3.5e-3 |

**Table 3.1: p-values for pairwise repeated measures t-tests of CM versus the other four error measures**

### 3.3.3 Experiment 2- Algorithm boundaries

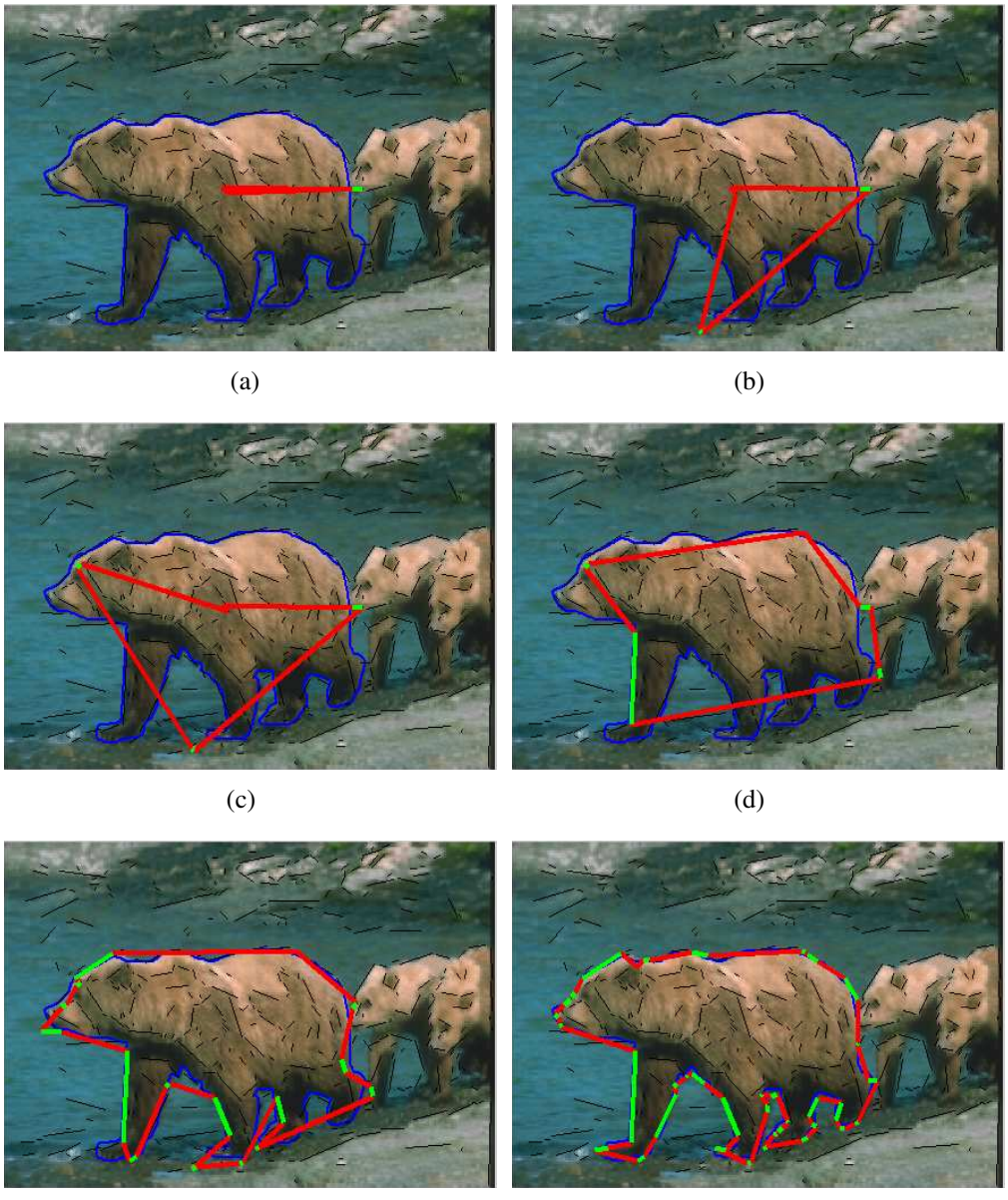
*Stimuli*: In Experiment 2, the reference shapes were again human segmentations drawn from the SOD database (Figure 3.9(b)). The two test shapes, however, were algorithm-generated boundaries. The *shape approximation algorithm* takes as input a set of line segments automatically detected in the image, as well as the hand-drawn reference shape. The goal of the algorithm is to estimate the cycle of line segments that best approximates the reference shape according to a specified error measure. The search begins by considering all possible pairs of line segments and evaluates the error of the quadrilateral they form. It then selects the minimum error quadrilateral and considers all updates to this cycle that involve insertion, deletion or replacement of a line segment, selecting one that reduces the approximation error. This process is repeated until the algorithm converges, i.e., all possible updates increase the error. See Figure 3.11.

We find that performance is improved if the search is probabilistic. Specifically, we rank the possible updates by the amount by which they reduce the error and select the update according to a probability model based upon the beta distribution. The aggressiveness of the algorithm in reducing the error is governed by a relaxation parameter $p^*$. Smaller values of $p^*$ result in a faster, more aggressive algorithm. However, we find that the algorithm converges to slightly lower error on average with less aggressive values (Figure 3.12). For the experiments here we selected a value of $p^* = 10^{-10}$, which converges in an average of about 45 iterations for the shapes we consider[1].

To generate candidate stimuli for our experiments, we ran this algorithm using each of our 5 shape similarity measures, on one human segmentation for each of the 30 SOD images.[2] A list of candidate stimulus triplets was thus produced. In order to ensure that the test shapes were neither too similar nor too different, they were selected to be

[1] Videos are available at `http://www.elderlab.yorku.ca/~vida/CM/POCV10_Vida_supp.zip` as supplementary material containing examples of this iterative shape approximation process.

[2] The search space seems to be smoother for RI and CM, resulting in more reliable convergence and lower errors.



(a) (b)

(c) (d)

(e) (f)

**Figure 3.11: Sample iterations of the Shape Approximation algorithm** - The shape approximation algorithm minimizes error by a probabilistic selection of insert, delete, or replace moves. Figures (a) to (f) show iterations 2, 3, 4, 8, 20, and 60 respectively.

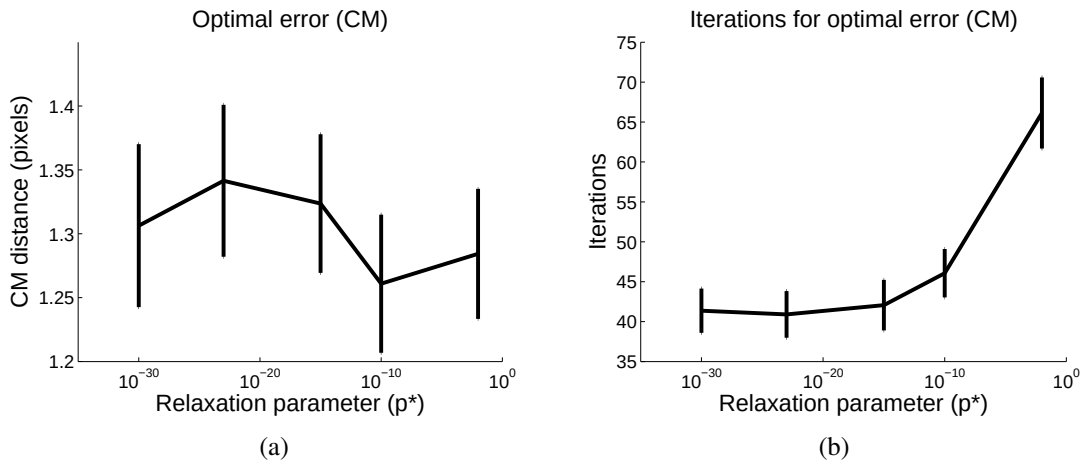



**Figure 3.12: Convergence of the Shape Approximation algorithm as a function of the relaxation parameter ($p^*$)** - (a) Mean error at convergence, and (b) Mean number of iterations to convergence. Error bars indicate standard error of the mean.

intermediate shapes generated 2-10 iterations apart in the same run of the algorithm, using a common shape similarity measure.

From this large set of candidate triplets, we again selected only those that generated disagreement between at least two of our measures. We then used a method similar to that employed for Experiment 1 to select test stimuli $B$ and $C$ that were maximally different. However, given that at intermediate stages the algorithm is capable of generating shapes that have very little in common with the target reference shapes, it was also important to limit dissimilarity between the test shapes and the reference shape in order to make the task meaningful for human subjects. To accomplish this, we selected triplets for which *dissimilarity* between test shapes $B$ and $C$ surpassed a threshold, and *similarity* between one of the test shapes $B$ and the reference shape $A$ surpassed a threshold. For each subject participating in the experiment, from this large subset of triplets 150 were randomly selected for each pairing of $CM$ with another competing measure $M$, with 20 percent overlap between pairs of subjects to allow estimation of inter-subject consistency, resulting in a total of 600 stimulus triplets shown to each subject. 9 subjects participated in the experiment.

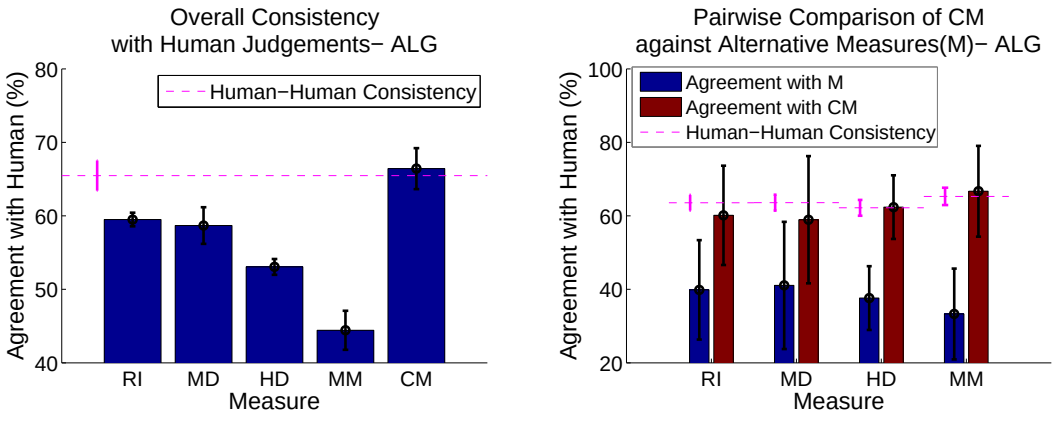



**Figure 3.13: Results of Experiment 2** - (a) Overall consistency with human subjects. (b) Consistency with human subjects for trials on which CM disagrees with each other measure $M \in \{RI, MD, HD, MM\}$. Error bars indicate standard error of the mean.

*Results*: Figure 3.13 shows the results of this experiment. We find again that the CM algorithm is most consistent with human judgements, and again is as predictive of human judgements as humans are of each other. We find that pairwise differences between CM and the other measures are statistically significant at the $p < .05$ level for the HD and MM measures, but not for the RI and MD measures (Table 3.1).

Interestingly, while the Hausdorff (HD) measure ranked second in Experiment 1, the Region Intersection (RI) and Mean Distance (MD) measures score better in Experiment 2. This shows how the appropriateness of some measures can vary with the stimuli. At the same time, the CM measure performs well in both cases, suggesting that it may generalize well.

## 3.4 Precision-Recall Analysis

Precision and recall measures [36] have also been widely used for evaluation of segmentation algorithms. For an algorithm boundary $A$ and a ground-truth boundary $B$, Precision is the proportion of boundary points on $A$ that are true positives: Precision $=$

$\frac{\text{Matched}(A,B)}{|A|}$, and Recall is the proportion of boundary points on $B$ that are actually detected: Recall $= \frac{\text{Matched}(B,A)}{|B|}$. High precision corresponds to a low false positive rate, whereas high recall corresponds to a low false negative (miss) rate. In order to calculate these measures, a method for matching points on the two boundaries is required.

In their original Precision-Recall approach (M-PR), Martin et al. [36] solved the correspondence problem as a minimum cost bipartite matching, where the cost of matching two points is proportional to the distance between them. A 1:1 matching is possible by adding outlier nodes. Any match to an outlier or beyond some distance threshold is counted as a mismatch. In a recent variation on this approach (E-PR), Estrada et al. [69] include 'no intervening contours' and 'same side' constraints. While these constraints serve to encourage ordering consistency between the two contours being matched, neither approach strictly enforces ordering consistency in a global sense.

We can assess the significance of the ordering constraint within the Precision-Recall framework by using the CM method of section 3.2.4 to match points, and pruning matches beyond a distance threshold. Among multiple matches incident on one point, only the one with the shortest distance between the matched points is preserved and the rest are pruned. Since the matching step is independent of the distance threshold, changing the distance threshold does not require re-computation of matchings as required by Estrada's method.

To evaluate each of these P-R measures against our human data, we ran each measure through our experiments (Section 3.3), selecting the test shape that yielded the highest F-measure on each trial, where $F = \frac{PR}{\alpha R+(1-\alpha)P}$ with $\alpha = 0.5$, varying the distance threshold from 1 to 13 pixels. The results in Figure 3.14 show that best agreement with human judgements is obtained using the CM matching method, suggesting that the global ordering constraint is still important within the Precision-Recall framework. Note that performance is best at high threshold values, indicating the importance of allowing quite distant matches. Note also that the CM matching method appears to

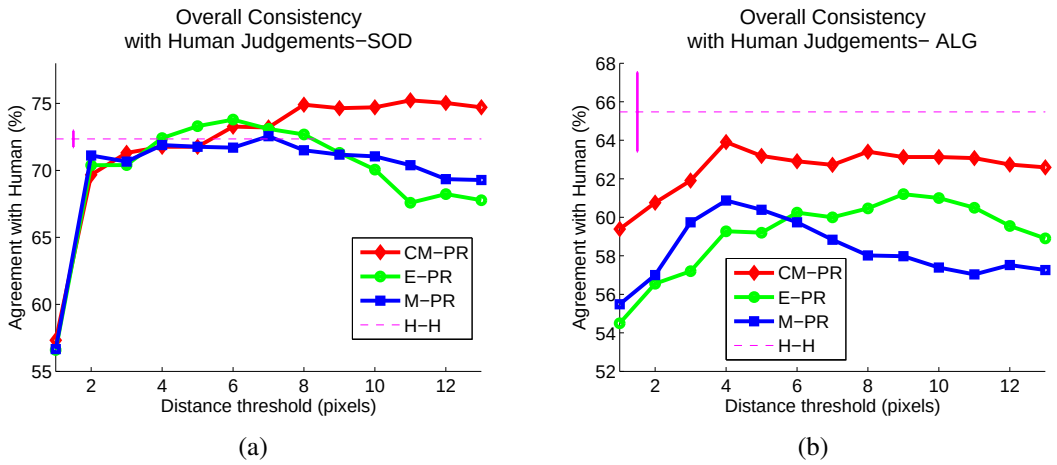



**Figure 3.14: Consistency between human subjects and error measures using precision-recall framework** - The dashed pink H-H line denotes human-human consistency. The vertical pink bar indicates standard error of the mean.

be relatively stable in consistency with human judgements once the distance threshold is sufficiently high, whereas the competing measures have a narrower 'sweet spot' at an intermediate threshold.

## 3.5 Conclusions and further considerations

Empirical performance evaluation of object segmentation algorithms requires a dataset of ground truth object segmentations and an appropriate error measure. In this chapter, we have introduced a dataset of ground truth object segmentations that can be used for this purpose. We then considered 5 error measures that have appeared in various forms in the literature, and analyzed their potential strengths and weaknesses. Finally, we psychophysically evaluated these measures using two distinct types of stimuli. Our results show that a Contour Mapping measure based upon contour bimorphisms between the boundaries of the object segmentations under comparison was most consistent with human judgements, and, amazingly, was as predictive of human judgements as human subjects were of each other. We also proposed using the same matching paradigm in

Precision-Recall analysis.

We believe that the perceptual consistency of the contour mapping measure derives from its sensitivity to prominent shape features that may have small area and its topological strictness, which requires that boundary points be considered as shapes rather than a scatter of points.

The region-based measure is often used as the evaluation measure in the contour grouping literature. Figure 3.15 compares this measure with our contour-based measure in the task of selecting the best contour among a set of algorithm contours. This figure shows that the best contour selected by the region-based error is not as good as the one that is selected by the contour-based error.

The following implementation details need to be taken into consideration:

- In the following chapters, the contour mapping measure is used as the main evaluation measure when needed. An exception is in cases where time complexity is an issue and approximations to the distance measure are acceptable. Calculation of the CM error for one pair of contours takes approximately 0.017 seconds on one core of an Intel i7 @3.40 GHz. However, as an example, evaluating 5000 contours per image against an average of 30 ground truth contours per image for 30 training images takes more than 21 hours.

- Note that the number of samples on longer ground truth contours is higher and this will lead to higher error values for larger objects. To remove this bias when combining evaluations over a set of objects with different sizes, we normalize the contour grouping measure by the square root of the area inside the ground truth contour, and will often refer to this dimensionless measure as **CMnorm**.

- There are some human errors or inconsistencies in selection of the salient objects in SOD. These cases are identified by comparing to the selections made by other subjects. The Mean of Max Consistency (MMC) for each object contour $O_j$

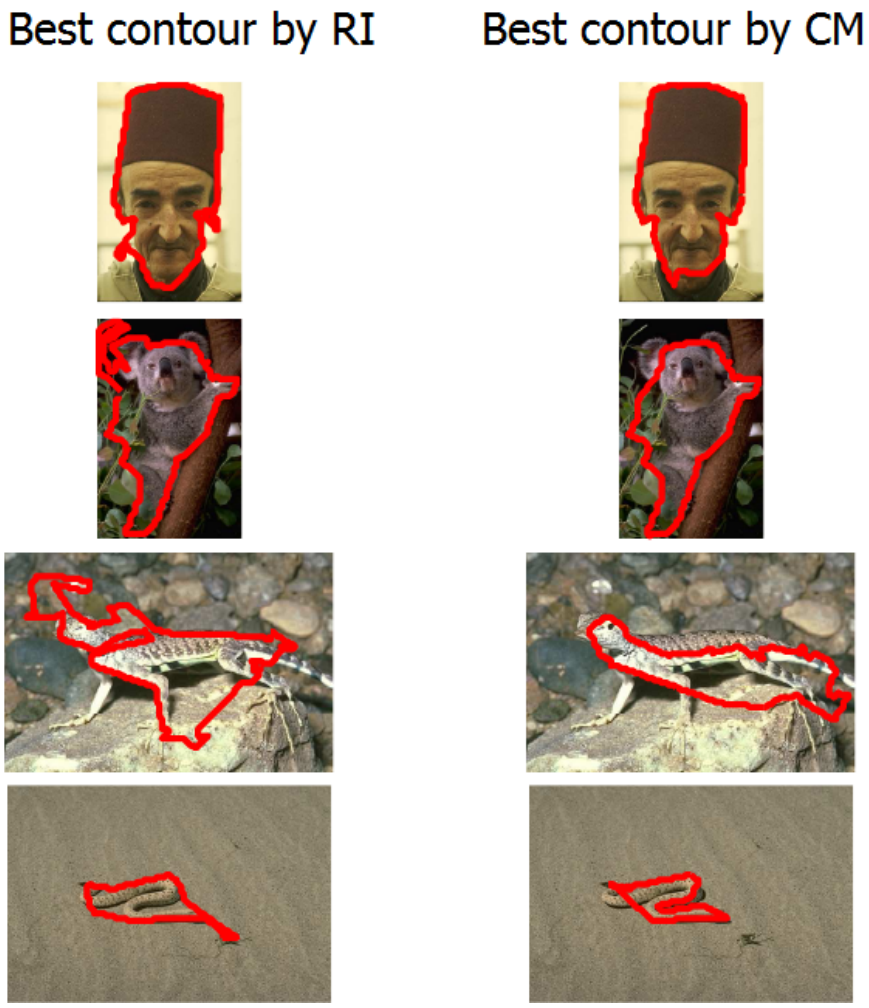



**Figure 3.15: Examples of differences between the region-based error measure RI and the contour-based error measure CM**- The region-based error is not sensitive to spikes, wiggles, and major shape differences. The contour-bassed criterion, however, produces segmentations that correspond better to human perception.

selected by subject $j$ is defined as

$$MMC(O_j) = \frac{1}{N-1} \sum_{i=1..N, i \neq j} \max_{O_i} \text{IoU}(O_j, O_i) \tag{3.13}$$

where $O_i$ are objects identified by human subject $i$, and $\text{IoU}(A, B)$ is the intersection over union measure as defined in 3.2.1. Objects are discarded as inconsistent if their $MMC$ values are lower than 0.6. This threshold is applied throughout the remainder of this thesis.

- To evaluate an algorithm-generated contour, it will be compared against all salient ground truth object boundaries generated by multiple human subjects for that image. The error of the contour is then defined as the *minimum* CMnorm error over all ground truth contours for that image.

# 4

# The Association Graph

Contour grouping methods search for the optimal cycle of local oriented primitives forming the boundary of a salient object (or objects) in the image. Edges or contour fragments approximating edges are often used as the local primitives. In this work, local oriented edges are first detected using an edge detection method. These edges are then grouped locally into subpixel-localized line segments of variable length. While discarding redundant information [6], these line segments provide explicit visual information that can be exploited by segmentation methods effectively. The goal is to group these line segments into ordered sequences forming the object boundary.

Line segments obtained in the image can be traversed from tip to tail, or tail to tip. Therefore, as in prior work [43], these line segments are duplicated to separately represent the two possible trajectories through each line segment. Each of the resulting segments forms a vertex in an association graph, and each edge in this graph represents a grouping hypothesis between specified endpoints of two segments. However, not all edges of the complete graph represent plausible groupings. Therefore, weak graph edges are pruned to obtain a sparse association graph.

In the following sections, I will first explain the preprocessing steps of edge and line detection. I will then discuss methods for forming the association graph. Parts of

this chapter have been published in [70].

## 4.1 Edge Detection

Edge detection is the first stage of dimensionality reduction. Elder [6] showed that information needed for higher level tasks, such as segmentation, is not discarded by edge detection.

How does the edge detection stage affect the grouping method, and which method should be used? There is a wide range of edge detectors available. It is not within the scope of this work to evaluate edge detection methods. I will therefore mention only a few edge detectors here, selected based on their performance or popularity:

- Canny edge detection[71]
- Pb (Probability of boundary) [36]
- gPb (global Pb) [72]
- Multi-scale edge detector of Elder and Zucker [73]

Sample edge maps by these methods are shown in Figure 4.1.

Edge detectors are often evaluated using a precision / recall framework. Ground truth edges are obtained from boundaries hand-drawn on a set of images by human subjects. The Berkeley Segmentation Dataset [3] contains ground truth segmentation boundaries, suitable for evaluating edge detection methods (see sample in Figure 4.1(b)). *Recall* is the fraction of ground truth edge pixels (edgels) that the edge detector is able to identify as edge points. *Precision*, on the other hand, is the fraction of algorithm edgels that are actually (ground truth) edgels.[1] Precision and recall values are often combined into F-measures [36]. If precision and recall are denoted as P and

[1] To tolerate small localization errors, correspondence of edgels is based on a distance threshold set as 1 percent of image diagonal size as suggested by Martin et al. [36], and within constraints set by Estrada and Jepson[69] for correct matching.

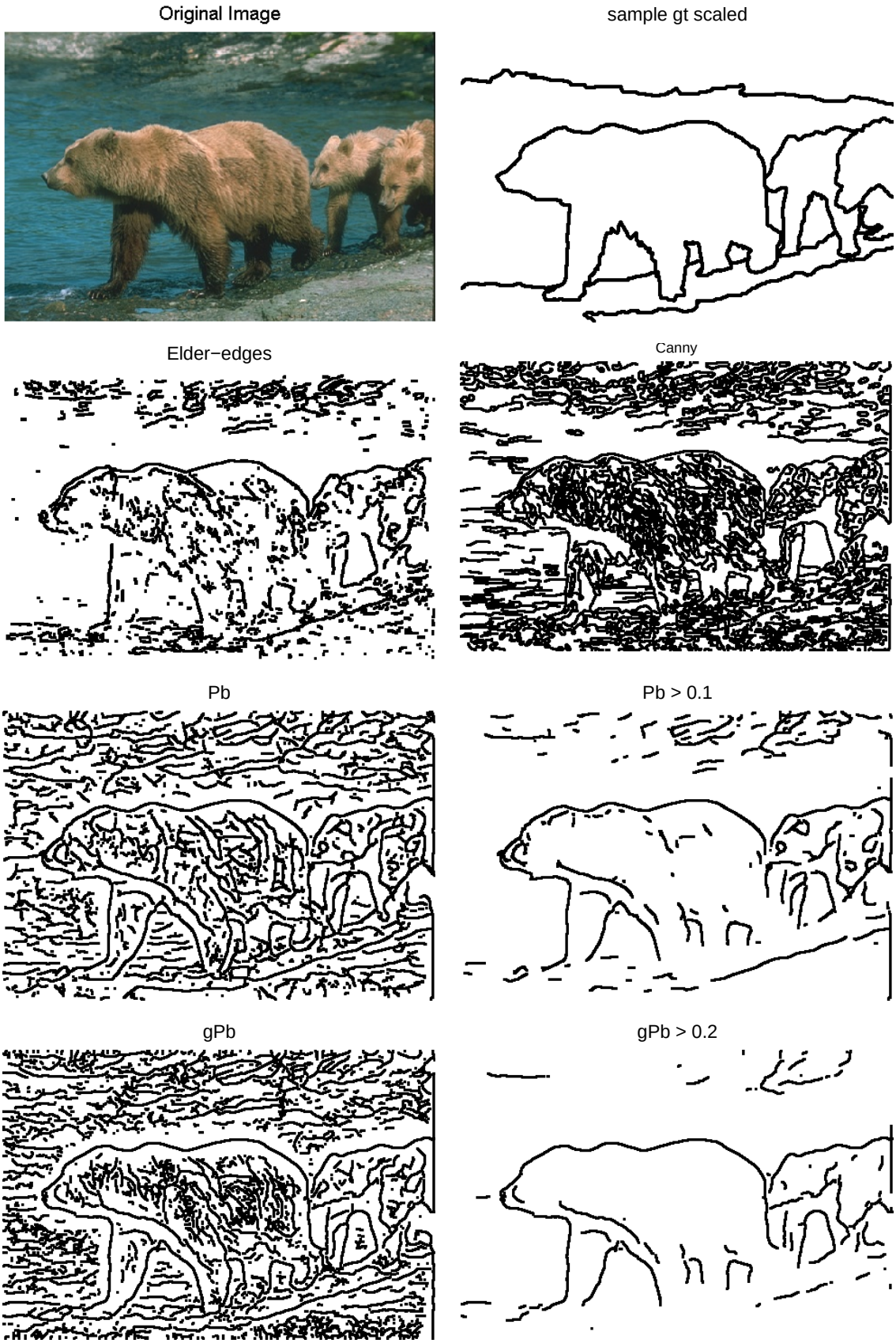



**Figure 4.1: Sample edge maps** - (a) Original image, (b) sample ground truth segmentation boundaries from BSD dataset [3] (traced by one subject), and edge maps by (c) Elder, (d) Canny, (e) Pb, (f) thresholded Pb, (g) gPb, and (h) thresholded gPb edge detectors. Parameters and thresholds are set as mentioned in the text.

R respectively,

$$F = PR/(\alpha R + (1-\alpha)P) \tag{4.1}$$

Often the weight of $\alpha = 0.5$ is used.

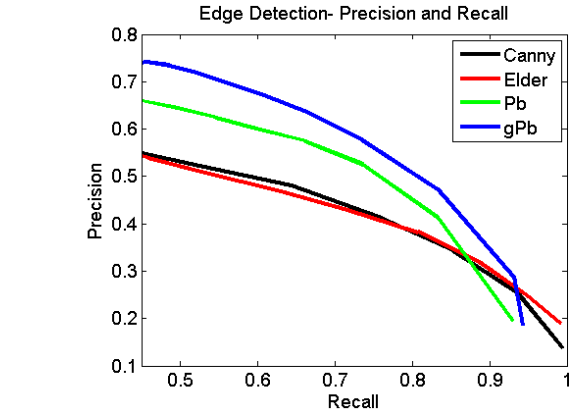
Figure 4.2(a) compares the above edge detectors based on their average precision and recall values over 30 training images.[1] As can be seen in this figure, there is a trade-off between precision and recall. Often high recall results in a higher percentage of false positives, and hence lower precision rates. The gPb method has the best precision-recall curve among the compared methods, as claimed by the authors [72].

The question is where on the precision and recall curve is the sweet spot for grouping algorithms. There are various strategies used in the literature: The suggestion in [72] is to maximize the F-measure for Pb and gPb.[2] The standard deviation of noise for the Elder and Zucker edge detector is learned from training image statistics, approximated over regions without edge information. These regions were selected by filtering out pixels belonging to Berkeley segmentation[3] boundaries.[3] For the Matlab implementation of the Canny edge detector, a threshold value of 0.7 times the highest edge magnitude in the image is suggested for a lower chance of disconnecting connected edgels. These settings are used in obtaining F-measures reported in Figure 4.2(b) and samples shown in Figure 4.1. However, as we will show in the next section, having the highest F-measure is not the best setting for grouping algorithms.

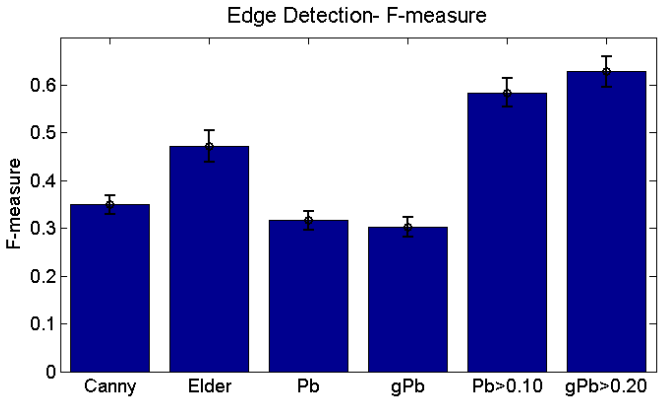
[1] The standard deviation of the Gaussian filters of the Canny edge detector was set to the default value of 1 (as implemented in Matlab R2010b). Default values are used for Pb and gPb as implemented by the authors. For Elder and Zucker edge detector, the standard deviation of noise parameter is varied. Minimum and maximum scales are set to 1 and 3 respectively. These values were selected by maximizing the F-measure.

[2] A threshold of 0.1 for Pb and 0.2 for gPb maximizes their F-measure value.

[3] The value of the standard deviation of noise learned from 30 training images is $\sigma_n = 8$.




(a)




(b)

**Figure 4.2: Comparison of edge detection methods** - on 30 training images at full resolution (a) by Precision and Recall measures, and (b) by F-measures with $\alpha = 0.5$. Error bars indicate standard error of the mean.

## 4.2 Line Approximation

Edge maps are not directly used in the contour grouping algorithm. They are first grouped locally into line segments. This initial grouping stage serves to reduce position and orientation noise in the local edges, and also reduces the size of the search space used in subsequent stages. However, even after this reduction of search space size, low precision of edge detectors can result in too many lines.

On the other hand, low recall values can lower the best performance achievable

by the grouping method. To see the effect of the preprocessing steps of edge and line detection, here we report an estimate of the minimum achievable error of contour grouping given the edge detection output studied in the previous section. Edges are grouped locally into subpixel-localized line segments of variable length using the line detection method proposed by Elder and Zucker [74]. In each iteration of this method, the longest line segment faithfully modeling a subset of a connected set of local edges is found. This subset is then removed from the original set. This process is repeated as long as line segments longer than a minimum length[1] can be fitted to the remaining local edges. The tangent direction of edges fitted by a line is kept within a range determined by a learned affine model.

Line map samples are shown in the left columns of Figures 4.3 and 4.4. These lines[2] are used in the Shape Approximation Algorithm discussed in Section 3.3.3 to approximate a ground truth boundary by minimizing the CM error. One ground truth boundary is selected per image from SOD among objects tagged as having the highest saliency and with maximum consistency among subjects (see for example Figure 4.3(b)). Sample groupings are shown in the right columns of Figures 4.3 and 4.4.

Figure 4.5(a) shows the average CM error of the best approximations found by the algorithm in 30 training images normalized by the square root of the area inside the ground truth contour. The first four edge detectors have a low achievable error, much lower than currently possible with fully automatic grouping algorithms. The number of lines, however, is different for these four methods, with the lowest number obtained given Elder & Zucker edge map 4.5(b). Thresholded Pb and gPb edge maps optimized to maximize F-measure can result in an even lower number of lines, yet these lines do not represent enough detail on the object boundary and therefore results in larger

[1]The minimum length parameter of the line detection algorithm is set to 5 pixels in full resolution images, 2 pixels for half resolution (scale 2), and 1 pixel in images with a quarter of the original image resolution (scale 3).

[2]To increase the convergence speed of the shape approximation algorithm, only line segments within 10 pixels of the ground truth boundary were included in the search. This threshold was set to 7 and 5 pixels for lower resolutions at scales 2, and 3 respectively.

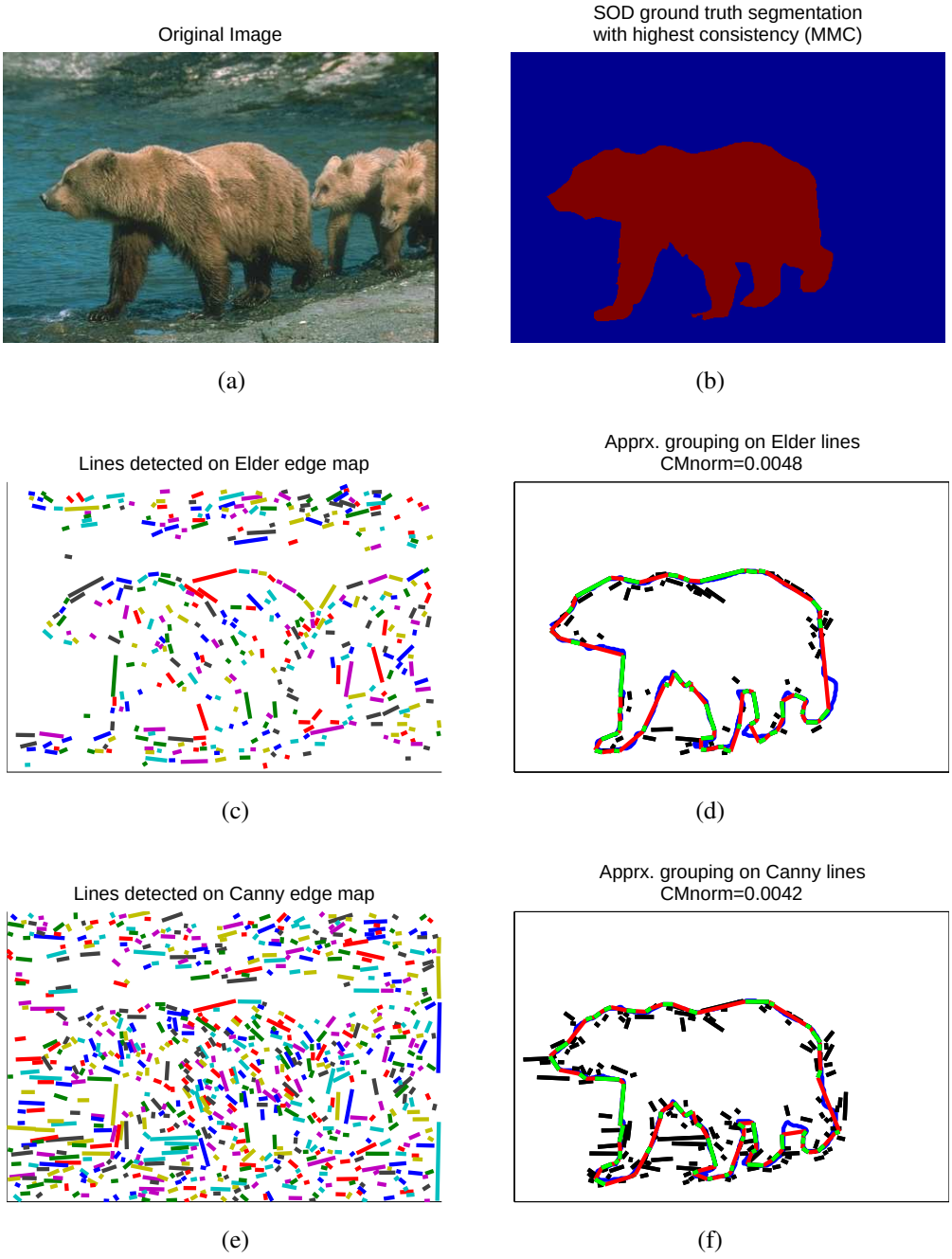



**Figure 4.3: Sample line maps and groupings** - (a) Original image and (b) SOD ground truth segmentation with maximum consistency among subjects. (c) Lines detected on Elder's edge map (randomly coloured), and d) grouping found by shape approximation algorithm to approximate the ground truth boundary shown in (b). The ground truth boundary is shown in blue. The approximating grouping is shown in alternating green and red, where green represents detected lines, linked by red virtual segments. Only line segments within the search range of the ground truth boundary (10 pixels) are shown. (e) and (f) show lines and groupings based on Canny edge map.

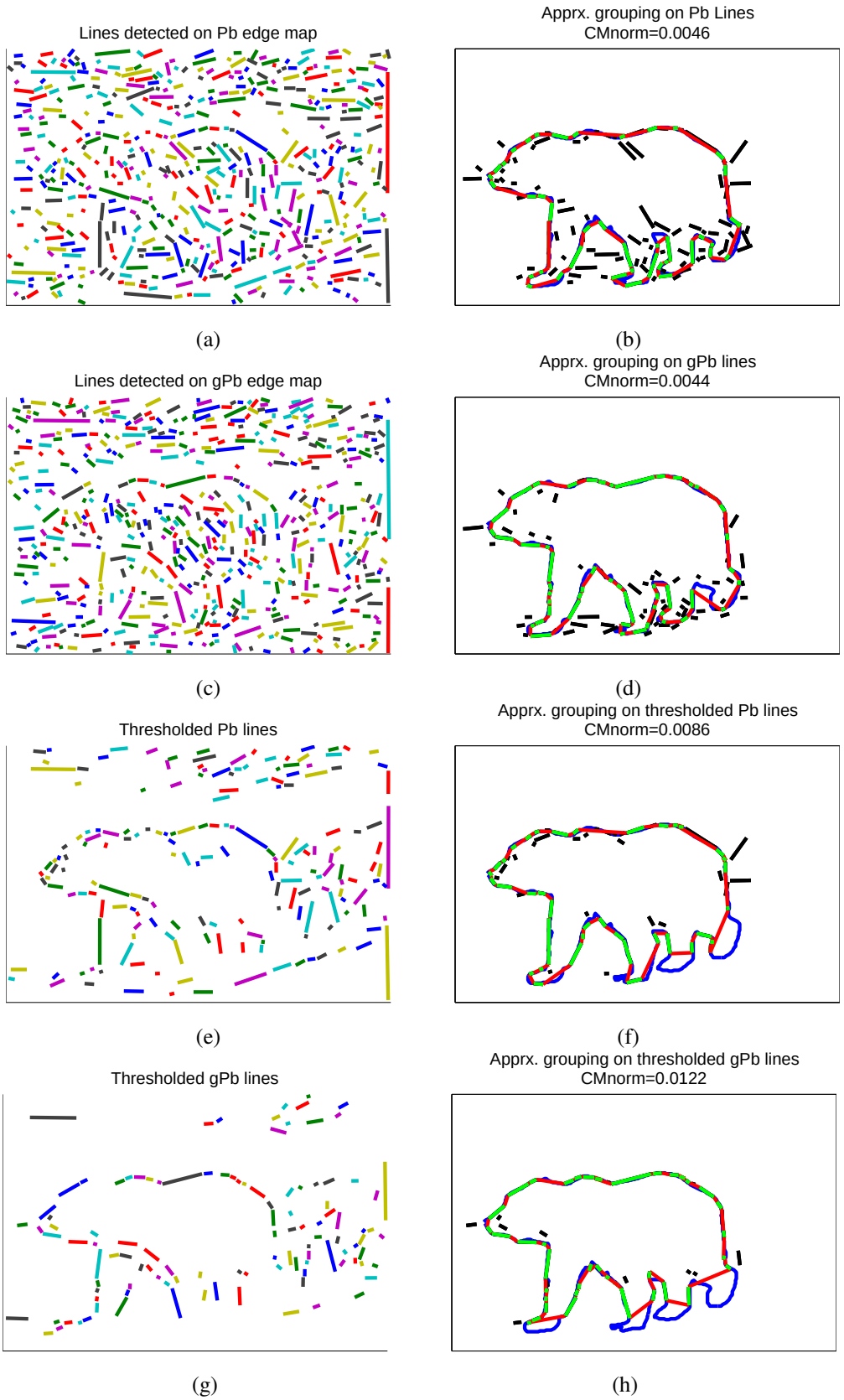



**Figure 4.4: Sample line maps and groupings- cont.** (a) and (b) Lines and groupings based on Pb map, (c) and (d) lines and groupings based on gPb map, (e) and (f) lines and groupings based on thresholded Pb map, (g) and (h) lines and groupings based on thresholded gPb map.

grouping errors. Through the rest of this dissertation, we will use Elder & Zucker edge detection to achieve relatively lower grouping error with relatively low number of lines.

Our goal is to use a multiscale framework in this dissertation, therefore we smooth and downsample images as suggested by [10]. There is less noise and clutter at low resolution and the search space is smaller. The effect on minimum achievable error and number of lines is shown in Figure 4.6. We will often refer to the full resolution as scale 1, to half resolution as scale 2, and to quarter resolution as scale 3, viewed as a resolution pyramid.[1] As this figure shows, there is a greater impact from going from scale 2 to 3 than going from scale 1 to 2, both in the number of lines and the error introduced. We will therefore start our contour grouping search at scale 3 and will then use the hypotheses to find a finer grouping at a higher resolution.

[1] A binomial filter is used to approximate the Gaussian filter for smoothing, as in [10].

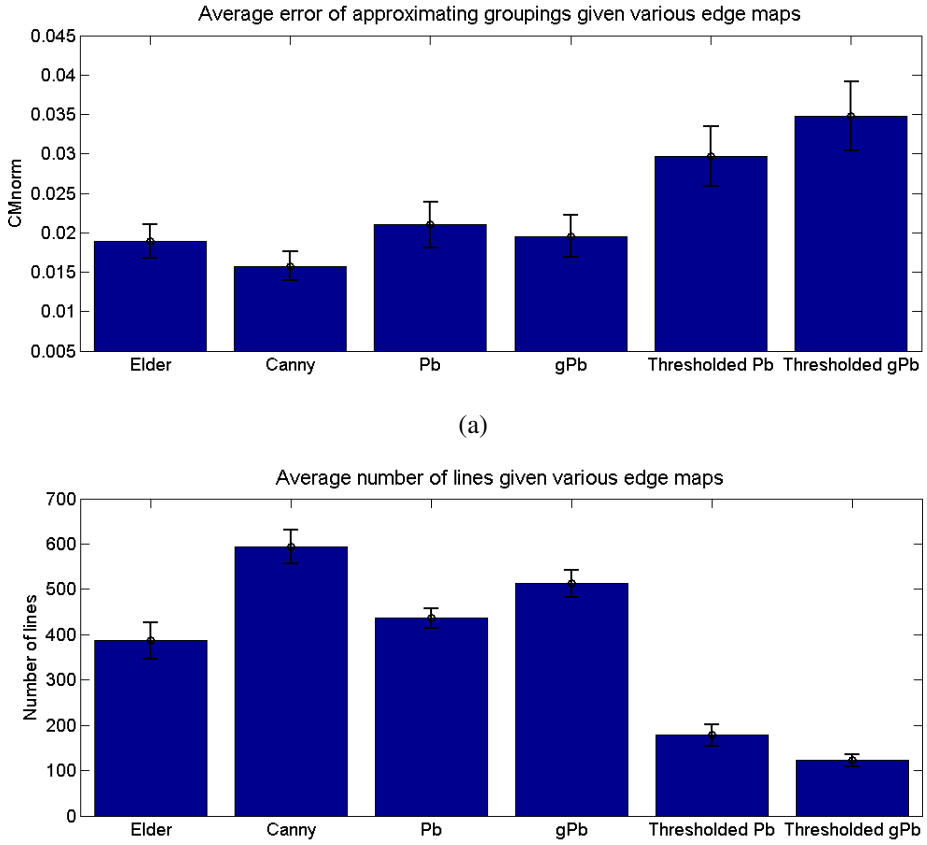



**Figure 4.5: Effect of edge maps on minimum achievable grouping error and complexity** - (a) Minimum achievable normalized CM error of approximating groupings, and (b) number of lines, given different edge maps and averaged over 30 training images.

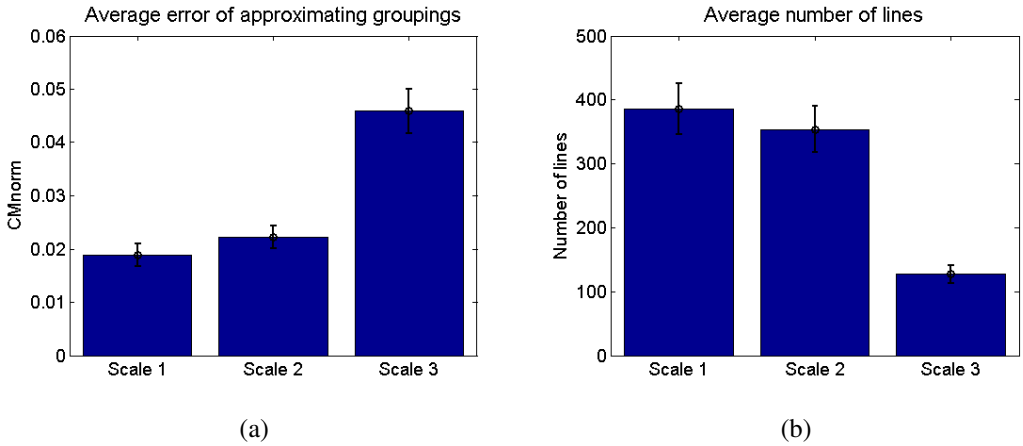



**Figure 4.6: Multi-scale line maps** - (a) Minimum achievable normalized CM error of approximating groupings, and (b) number of lines, given Elder and Zucker edge maps at different resolution levels (scales 1 to 3), averaged over 30 training images.

## 4.3 Forming the Association Graph

The first stage of processing in our method extracts from the image a directed graph representation of the oriented structure in the image. As in prior work (e.g. [43]), the line segments obtained by line approximation (as explained in the previous section) are duplicated to separately represent the two possible trajectories through each line segment. Each of the resulting segments forms a vertex in our association graph, and each edge in this directed graph represents a grouping hypothesis from a specified endpoint of one segment to a specified endpoint of another segment. For $N$ line segments in the image, there will be $2N$ nodes and $4N(N-1)$ directed links in the graph. Search in the complete graph is infeasible. Moreover, for any particular segment typically only a small number of segments in the image form plausible continuations, therefore the grouping graph is often pruned to lower the graph search complexity by limiting the number of links incident on each node to $k << N$. This pruning results in a sparse directed graph with fixed maximal out-degree $k$, limiting the size of the graph to $|E| \in \mathcal{O}(kN)$.

In order to learn the statistics required to group these segments, we conducted a hand-labelling exercise on the segments extracted for our training images. For each training image, the ground truth boundary with the highest Mean of Max Consistency (MMC), as defined in Equation 3.13, was selected from SOD. Then 3 subjects selected the cycle of segments that they felt best approximated the object boundary. We will refer to these ground truth cycles as the HND dataset (see Figure 4.7).

We used these ground truth cycles to learn models for local association between segments. In particular, at inference we wish to assign to each edge in our graph a weight equal to the log likelihood ratio of the geometric and photometric relationship between the two segments, conditioned on whether they are or are not neighbours on a ground truth cycle [5, 9, 10] (referred to as the ON and OFF conditions in the sequel). Statistics for the ON condition are learned from neighbouring segments in our ground

truth cycles, while the OFF statistics are learned from randomly selected segment pairs.

The relational cues for a link between an endpoint of segment $i$ and an endpoint of segment $j$ (see Figure 4.8) include [5]:

1. Proximity $r_{ij}$, i.e., the distance or gap between two detected line segments $i$ and $j$,
2. Parallelism ($\vec{\theta}_{ij} + \overleftarrow{\theta}_{ji}$), measured as the sum of the two angles formed by the linear interpolant connecting the segments,
3. Cocircularity ($\vec{\theta}_{ij} - \overleftarrow{\theta}_{ji}$), measured as the difference of these two angles,
4. Brightness difference ($(I_{i1} + I_{i2})/2 - (I_{j1} + I_{j2})/2$), measured as the difference between the mean luminance at the two segments.

I will use two different approaches to model the above cues. Similar to prior methods [5, 9, 10], I will first assume that the above cues are independent conditioned on being ON or OFF, and will model the marginals independently (naive Bayes). I will then have a second approach in which I will learn a Gaussian mixture model for each of the ON and OFF conditions. This will allow small dependencies between the cues to be captured. The performance of these models will then be compared and evaluated based on the following criteria:

1. **Maintaining ground truth links:** It is desirable for the graph construction method to maintain the ground truth links in the graph. Therefore the probabilistic model can be evaluated based on the the miss rate, defined as the percentage of ground truth links missing in the pruned graph.

   Note that a link between two segment endpoints represented by vertices $i$ and $j$ in the sparse graph is considered missing only if neither edge $(i, j)$ nor edge $(j, i)$ are present in the graph.

2. **Ranking of ground truth links:** The model should give high ranks to ground truth links relative to other links in the graph.

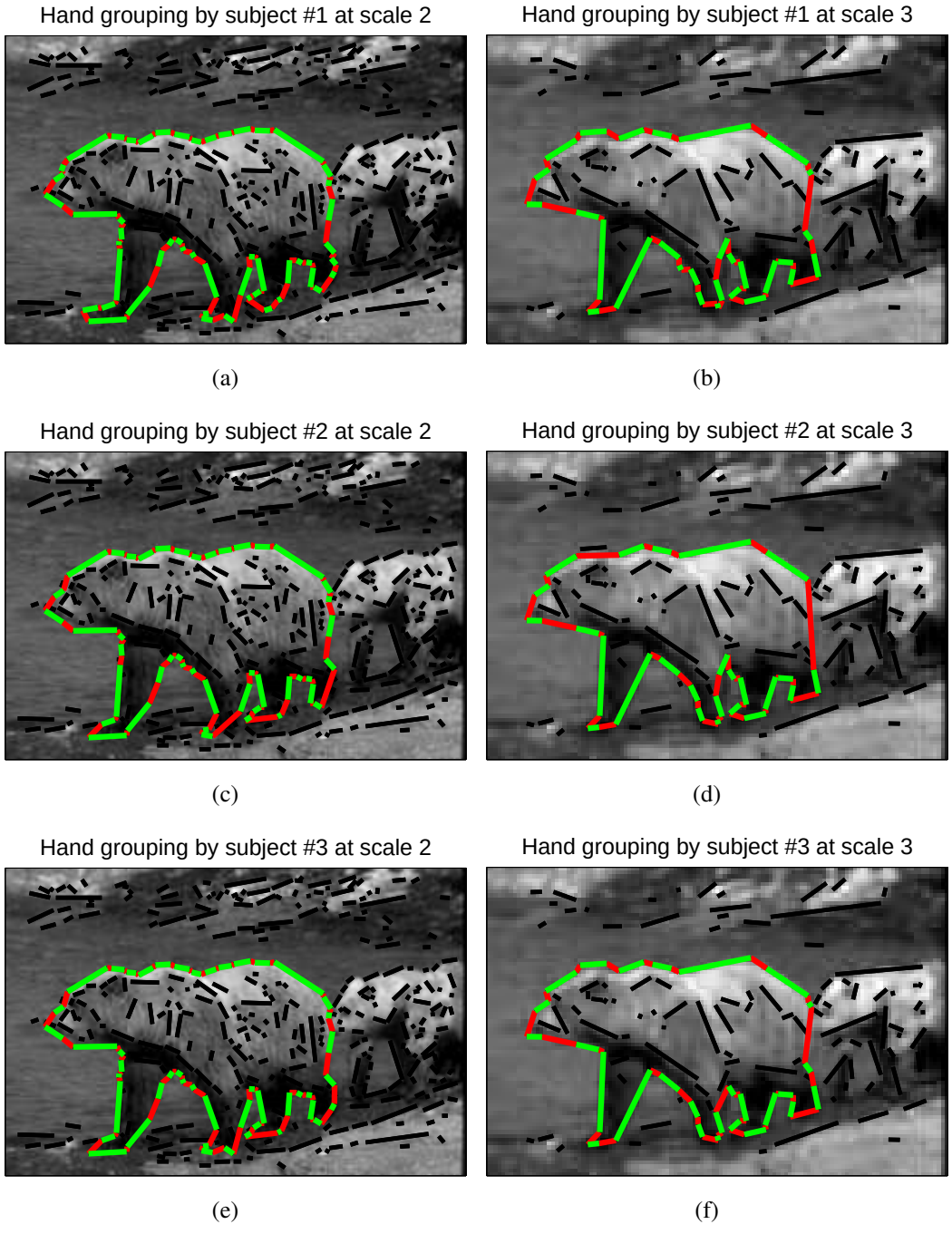



**Figure 4.7: Samples ground truth cycles in HND dataset** - Three subjects have selected the cycle of segments that they felt best approximated the object boundary in HND. Ground truth cycles for one sample image are shown at two scales.

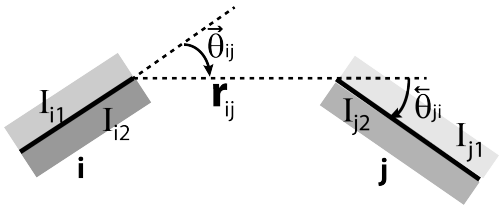



**Figure 4.8: Local cues** - Please refer to the text for details. (Reproduced from [5]).

3. **Minimum achievable error:** Since our final goal is to implement a salient object segmentation algorithm, we need to minimize the error introduced in the graph construction stage. This can be measured by the minimum achievable error in the graph.

4. **Modeling data:** In addition to the structure of the graph, the weights associated with the links in the graph are critical. These values will steer the process of generating contour hypotheses and determine the ranking of the contours. In a probabilistic approach, it is important that these weight are obtained from a model that best fits the distribution of data.

### 4.3.1 Method I: Independent Binary Cues

Assuming independence of cues conditioned on links being ON or OFF simplifies the models:

$$L_{ij} = \prod_{k=1}^{K} L_{ij}^{k} = \prod_{k=1}^{K} \frac{p(d_{ij}^{k}|\mathrm{ON})}{p(d_{ij}^{k}|\mathrm{OFF})} \tag{4.2}$$

In the following sections, the models for each cue will be learned independently given training data. At the time of inference, each cue $k$ of the four binary cues will contribute a likelihood ratio $L_{ij}^{k}$ to the product in the above equation.

#### 4.3.1.1 Proximity

Proximity is an important cue in grouping. Elder and Goldberg [5] showed that the inferential power of the proximity cue is much higher than other binary cues, reducing the entropy in the grouping decision by more than 70% in their dataset.

We will use two approaches for modeling the proximity cue:

1. Parametric: The ON and OFF distributions are modeled by parametric models such as the generalized Laplace or the log normal distribution models. The likelihood ratio value is then obtained from the ON and OFF distributions at the time of inference.

2. Non-parametric: We model the likelihood ratio directly using a histogram, and then fit this histogram.

Figure 4.9(a) and (b) show parametric models fitted to the data using maximum likelihood estimation. Figure 4.9(c) shows a linear and a piecewise linear model fitted directly to the likelihood ratio values. The plots show that the piecewise linear model is a better fit to the data and will be used as our proximity cue model.

The log normal probability distribution function is defined as:[1]

$$p(x|\mu,\sigma) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-\frac{(\log(x)-\mu)^2}{2\sigma^2}} \tag{4.3}$$

The generalized Laplace probability distribution function has the following form:[2]

$$p(x|\mu,\sigma,\gamma) = Ae^{-(c|x-\mu|/\sigma)^\gamma} \tag{4.4}$$

$$c = \sqrt{\frac{\Gamma(3/\gamma)}{\Gamma(1/\gamma)}}$$

$$A = \frac{c\gamma}{2\sigma\Gamma(1/\gamma)}$$

[1]The maximum likelihood estimates for the parameters of log normal distributions are $\mu = 1.17$ and $\sigma = 0.82$ for the ON gap data and $\mu = 3.56$ and $\sigma = 0.69$ for the OFF proximity data at scale 3.

[2]The maximum likelihood estimates for the parameters of the generalized Laplace distribution are $\mu = 44.27$ pixels, $\sigma = 23.18$ pixels and $\gamma = 2.79$ for the OFF proximity data at scale 3.

The linear model in log-log space is defined as:[1]

$$\log(\hat{L}) = a + b\ \log(\text{gap}) \tag{4.5}$$

where $\hat{L}$ is the estimated likelihood ratio. The piecewise linear model in the log-log space is similarly defined as:[2]

$$\log(\hat{L}) = \begin{cases} a_1 + b_1 \log(\text{gap}), & \text{if gap} <= g_0; \\ a_2 + b_2 \log(\text{gap}), & \text{if gap} >= g_0; \end{cases} \tag{4.6}$$

#### 4.3.1.2 Parallelism

The parallelism cue together with the cocircularity cue represents the Gestalt law of good continuation. Similar to the proximity cue, we take two approaches to modelling the ON and OFF distributions separately to calculate the likelihood ratio values, and also directly model the likelihood ratio histograms. Figure 4.10 (a) and (b) show the generalized Laplace model for ON data[3] and the triangular distribution for OFF data[4] defined as

$$p(x|a, b) = a + b\,\text{abs}(x) \tag{4.7}$$

This theoretically derived [5] distribution models the parallelism cue for the OFF condition based on the assumption that segment orientation is uniformly distributed. Figure 4.10 (c) shows the model obtained from the above ON and OFF distributions, as well as a direct model based on a mixture of distributions:[5]

$$f(x|A, \mu, \sigma, \gamma) = A(e^{-(|x-\mu|/\sigma)^\gamma} + e^{-(|x+\mu|/\sigma)^\gamma}) \tag{4.8}$$

[1]The least squared error estimates for the parameters of the linear model of the likelihood ratio of proximity at scale 3 are $a = 5.80$ and $b = -2.09$.

[2]The least squared error estimates for the parameters of the piecewise linear model of the likelihood ratio of proximity at scale 3 are $a_1 = 4.92$, $b_1 = 0.17$, $a_2 = 6.54$, $b_2 = -2.51$ and $g_0 = 5.22$.

[3]Assuming a symmetric distributions at $\mu = 0$ degrees, the maximum likelihood estimates for the parameters of the generalized Laplace model of the ON parallelism data (measured in degrees) at scale 3 are $\sigma = 76.49$ degrees, and $\gamma = 1.56$.

[4]Assuming a symmetric distribution, the parameter values for the triangular distribution are $a = 1/360$ deg$^{-1}$ and $b = 1/(360)^2$ deg$^{-2}$.

[5]The least squared error estimates for the parameters of the double Laplace model of the likelihood ratio of parallelism (measured in degrees) at scale 3 are $A = 1.15$, $\mu = 26.76$ degrees, $\sigma = 97.99$ degrees, and $\gamma = 1.30$.

The plots show that this double Laplace model of the likelihood ratio is a better fit to the data and will be used as our parallelism cue model.

#### 4.3.1.3 Cocircularity

As mentioned in [5], the estimation of cocircularity cue is ill conditioned when the separation between segments is small. The standard deviation of the cocircularity cue is higher for small gaps and hence the cue is weaker for grouping of segments that are close to each other.

For this reason, the ON data was separately modeled for gaps $<= 1$ pixel and gaps $> 1$ pixel as shown in Figure 4.11.[1] The OFF data is modeled with the same triangular model used for the parallelism cue.

#### 4.3.1.4 Brightness

The brightness difference cue in ON and OFF conditions is also modelled using the generalized Laplace model as shown in Figure 4.12.[2] Note that the brightness values are first normalized to the $[0, 1]$ range, and therefore the cue values will lie between -1 and +1.

[1] Assuming symmetric distributions at $\mu = 0$ degrees, the maximum likelihood estimates for the parameters of the generalized Laplace model of the ON cocircularity data (measured in degrees) at scale 3 are $\sigma = 97.15$, and $\gamma = 3.28$ when gap is less than 1 pixel and $\sigma = 91.31$, and $\gamma = 2.20$ when gap is larger than 1 pixel.

[2] Assuming symmetric distributions at $\mu = 0$, the maximum likelihood estimates for the parameters of the generalized Laplace model are $\sigma = 0.11$, and $\gamma = 1.06$ for the ON brightness data and $\sigma = 0.18$, and $\gamma = 1.61$ for the OFF brightness data at scale 3.

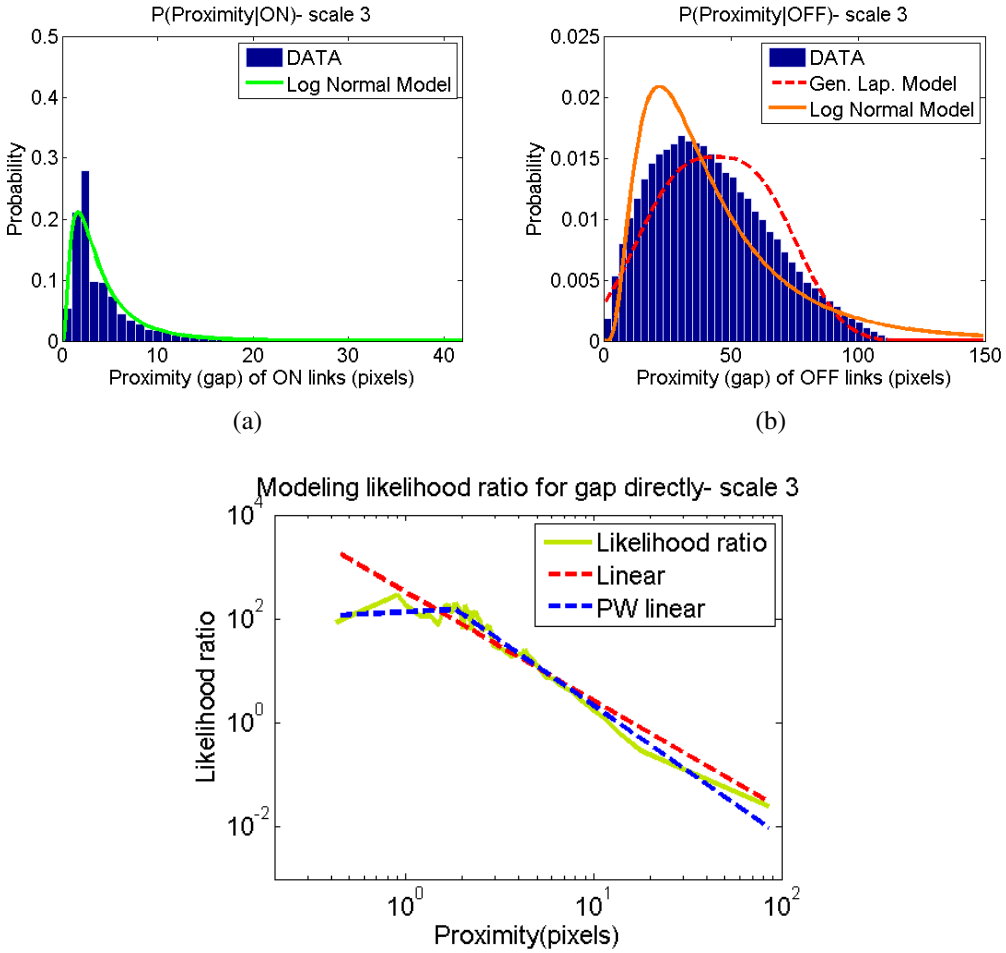
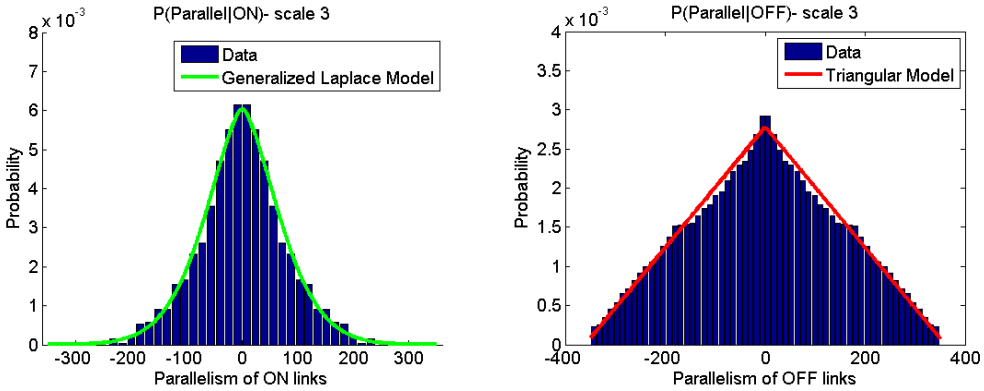
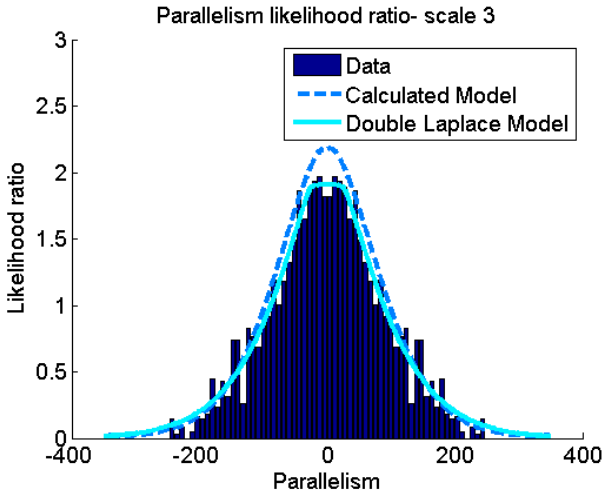



**Figure 4.9: Models for the proximity cue at scale 3** - (a) Log normal model for the ON distribution, (b) log normal and generalized Laplace distribution models for the OFF distribution, and (c) a linear and a piecewise linear model fitted directly to the likelihood ratio values obtained from histograms of ON and OFF data.




(a) (b)




(c)

**Figure 4.10: Models for the parallelism cue at scale 3** - (a) Generalized Laplace model for the ON distribution, (b) triangular distribution model for the OFF distribution, and (c) the calculated likelihood ratio model based on the ON and OFF distributions in (a) and (b) and a Laplace mixture model fitted directly to the likelihood ratio values obtained from histograms of ON and OFF data.

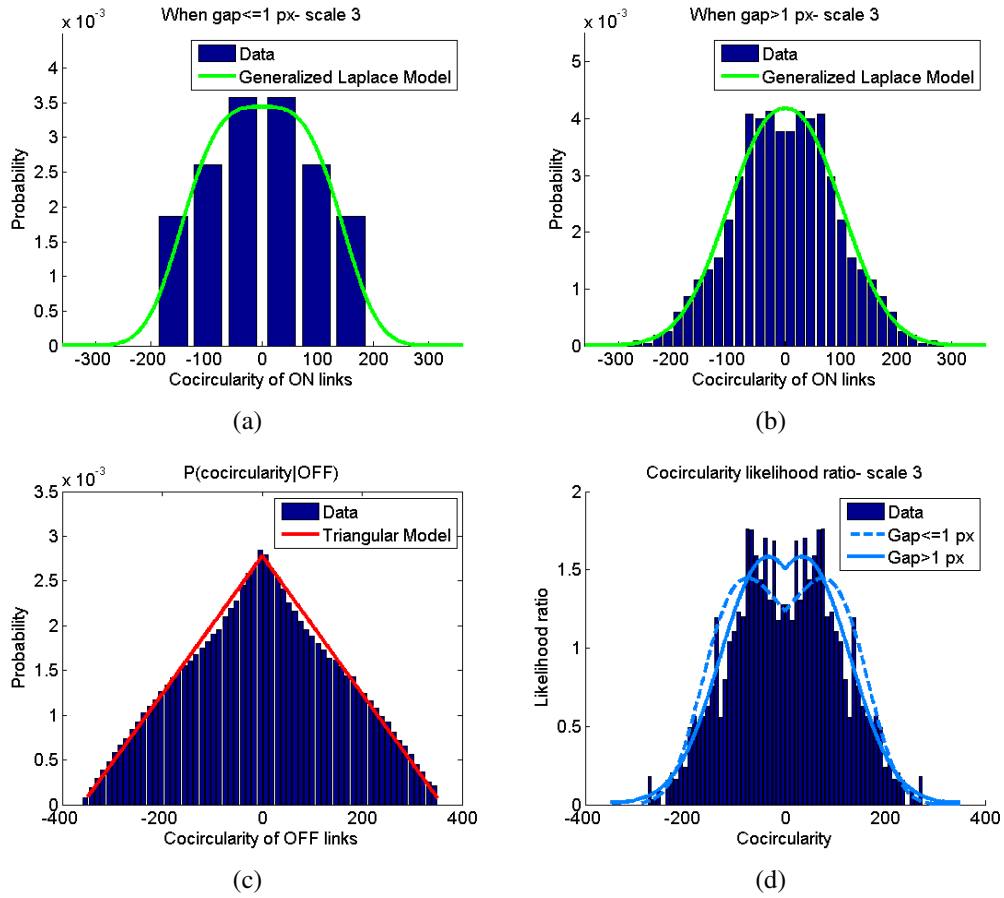
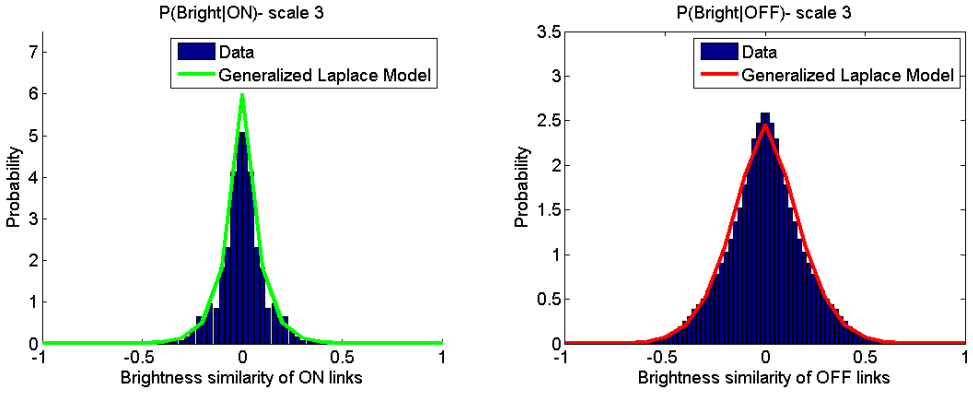
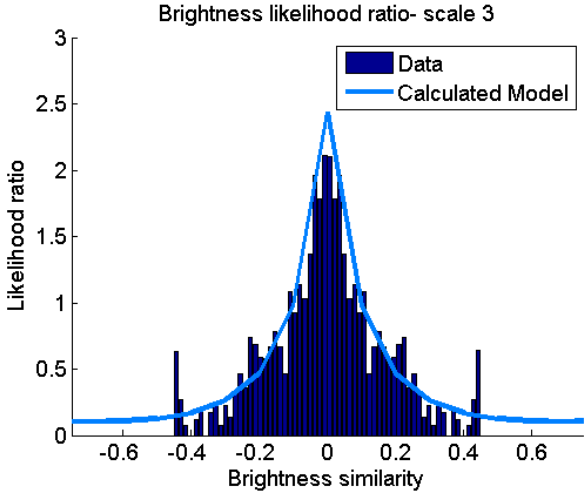



**Figure 4.11: Models for the cocircularity cue at scale 3** - (a) Generalized Laplace model for the ON distribution when gap is less than 1 pixel, (b) generalized Laplace model for the ON distribution when gap is more than 1 pixel, (c) triangular distribution model for the OFF distribution, and (d) the resulting likelihood ratio models.




(a) (b)




(c)

**Figure 4.12: Models for the brightness cue at scale 3** - Generalized Laplace model for the (a) ON and (b) OFF distributions, and (c) the calculated likelihood ratio model.

### 4.3.2 Method II: Dependent Binary Cues

The assumption that binary cues are independent is not entirely correct. For example we saw that the distribution of the cocircularity cue is different for different range of gap values. A multivariate model can help in exploring the dependencies between the cues and the effect on the construction of the sparse graph.

Here we employ a multivariate Gaussian mixture model (GMM) [75]. The ON and OFF distributions can be modeled as:

$$p(d_{ij}|\{i,j\} \in \text{ON}) = \sum_{c=1}^{C_{\text{ON}}} \pi_c \mathcal{N}(d_{ij}|\mu_c^{\text{ON}}, \Sigma_c^{\text{ON}}) \tag{4.9}$$

$$p(d_{ij}|\{i,j\} \in \text{OFF}) = \sum_{c=1}^{C_{\text{OFF}}} \pi_c \mathcal{N}(d_{ij}|\mu_c^{\text{OFF}}, \Sigma_c^{\text{OFF}}) \tag{4.10}$$

$$L_{ij} = \frac{p(d_{ij}|\{i,j\} \in \text{ON})}{p(d_{ij}|\{i,j\} \in \text{OFF})} \tag{4.11}$$

where $C_{\text{ON}}$ and $C_{\text{OFF}}$ are the number of components for ON and OFF data respectively. $\mu_c^{\text{ON}}$ and $\Sigma_c^{\text{ON}}$ are the mean vector and the covariance matrix for the $c^{\text{th}}$ component of the ON mixture model, while $\mu_c^{\text{OFF}}$ and $\Sigma_c^{\text{OFF}}$ are similarly defined for the OFF mixture model.

Each data point is a 4-dimensional vector of the local binary cues. The cues used here are proximity (log of gap), parallelism, cocircularity, and brightness difference. Given the number of components of the Gaussian mixture, the parameters can be optimized using the *expectation maximization* (EM) algorithm [75]. In addition to whitening[1] [75], we also use diagonal covariances for the distributions for better convergence of the EM algorithm. Increasing the number of components will improve the fit to training data, but over-fitting must be avoided. Figures 4.13(a) and (b) show the av-

[1]Whitening is done by standardization and rotation of data, so it has zero mean and unit covariance. This is done separately for ON and OFF data. Whitening the data helps in convergence of the EM algorithm and avoidance of ill-conditioned covariances, as suggested in Matlab. At inference, these learned transformations are applied to the data before computing the likelihoods in equation 4.11.

erage negative log likelihood using 6-fold cross-validation on our training dataset of 30 images given various values for the number of components for the ON and OFF distributions. Although the average negative log likelihood reaches a plateau at around 8 components, it does not converge for OFF data, and EM is computationally expensive at higher number of components[1]. We therefore optimized the number of mixture components by minimizing the average number of edges in the ground truth cycles not represented in the sparse graph, using 6-fold cross-validation on our training dataset of 30 images. Figures (c) and (d) show the miss rate across various values for the number of components of the ON distribution averaged over the number of components for the OFF distribution, and vice versa[2]. Figures (c) is very noisy suggesting insufficient data. We optimized the number of mixture components to 12 components for ON distribution and 9 components for the OFF distribution in scale 3.[3]

## 4.4 Evaluation of the graph construction methods

The sparse graph is formed by deleting all but the $k$ outgoing edges with highest weight $w_{ij} = \log(L_{ij})$ for each vertex $i$ (we use $k = 20$). Note that weights are symmetric, $w_{ij} = w_{ji}$, but due to the quota on out-degree, the presence of edge $\{i, j\}$ does not necessarily imply the presence of edge $\{j, i\}$.

Graph construction models are compared based on 3 criteria:

1. The average miss rate of ground truth links in the graph over 30 training images. This is an indication of learning performance. Figure 4.14(a) shows the average

[1]For each value $k$ for the number of components, EM is repeated 5 times with $k$ observations selected randomly from the data as initial component means. The initial covariance matrices for all components is set to a diagonal matrix, with the element $j$ on the diagonal set to the variance of the $j^{\text{th}}$ cue. The solution with the largest likelihood among the 5 repetitions is returned.

[2]The miss rate map is very noisy, therefore average values were used to optimize the number of components.

[3]In scale 2, the optimal number of components based on cross validation are 11 for the ON distribution and 10 for the OFF distribution.

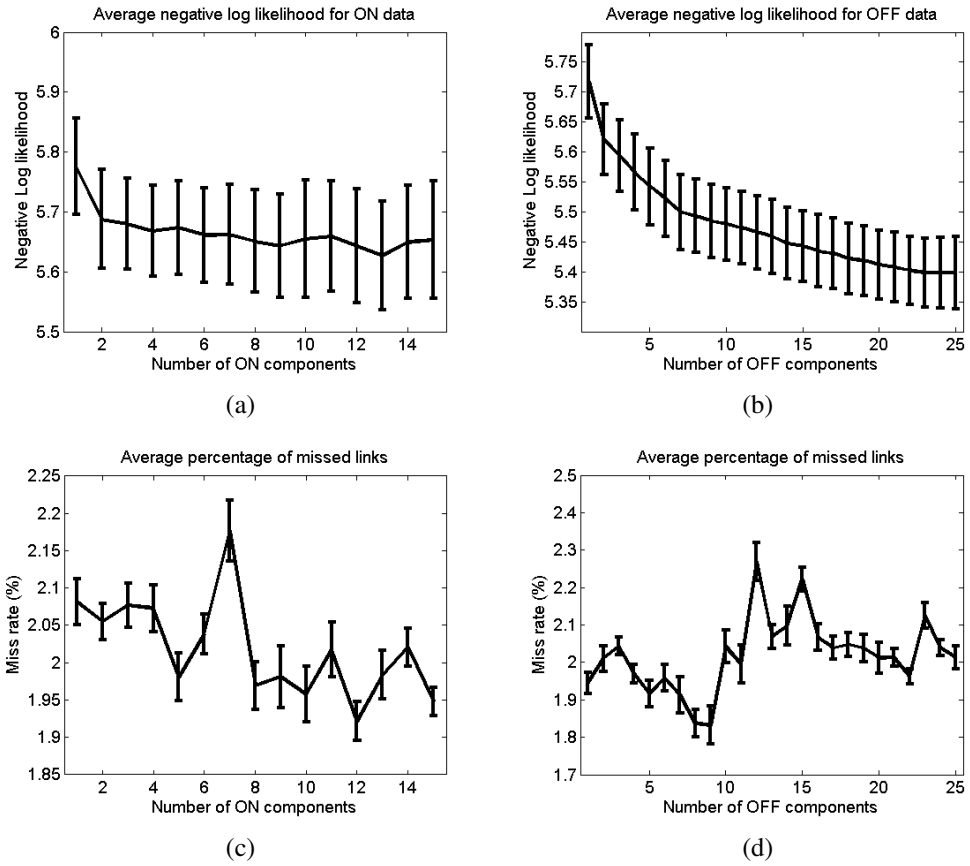



**Figure 4.13: Cross validation results given the number of components for the mixture models with diagonal covariances**- (a) and (b) Average negative log likelihood, and (c) and (d) average miss rate, among 6-fold cross validation given various number of components for the ON and the OFF distributions. Error bars indicate standard error of the mean.

miss rate of independent models versus the mixture model. Assuming a diagonal covariance for the mixture models improves learning. Note that this is not equivalent to assuming independence of the cues, since there is still some interaction being modelled by the mixture model.[1]

2. The average rank of the ground truth links among the links incident on an adjacent node over 30 training images. The 3 models compared above have almost the same average rank (see Figure 4.14(b)).

[1]Miss rate is slightly higher if each cue is modelled with a mixture of Gaussians independently.

3. The average CMnorm error of approximating groupings on 40 validation images as shown in Figure 4.14(c). The approximating grouping is found in each image by a modified Shape Approximation Algorithm minimizing the error with respect to a ground truth boundary (section 4.2), constraining the cycle to be in the sparse graph.

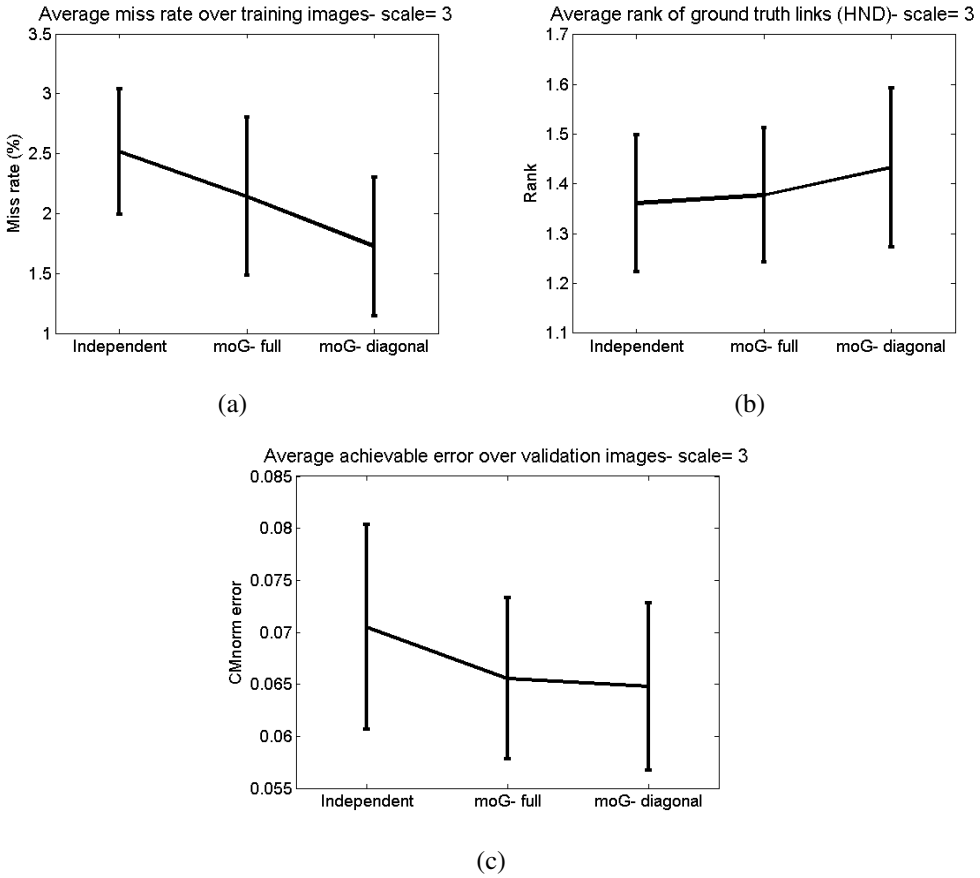



**Figure 4.14: Comparison of graph construction models**- Comparing independent models and Gaussian mixture models with full and diagonal covariances based on (a) average miss rate and (b) average rank of ground truth links over training images; and (c) average achievable CMnorm errors of approximate groupings in graph among validation images. Error bars indicate standard error of the mean.

The mixture model with diagonal covariances leads to significantly lower miss rate, as indicated by the p-value at $\alpha < 0.05$ level shown in Table 4.1, and slightly lower

| Measure | Indep. vs. moG-full | Indep. vs. moG-diagonal |
|---|---|---|
| Miss rate (training set) | 0.4044 | **0.0410** |
| Rank (training set) | 0.8080 | 0.4076 |
| CMnorm error (validation set) | 0.4089 | 0.3063 |

**Table 4.1: p-values for pairwise repeated measures t-tests of graph construction methods**- Comparing independent models and Gaussian mixture models with full and diagonal covariances based on p-values for pairwise repeated measures t-tests done on measures of Figure 4.14. The difference in miss rates using independent versus Gaussian mixtures models with diagonal covariances is statistically significant at $\alpha < 0.05$ level. All other differences between different models for graph construction are not statistically significant at $\alpha < 0.05$ level.

achievable error in the graph compared with independent models for each binary cue. This is also true in higher resolution images.[1] We will therefore use the mixture models with diagonal covariances in the subsequent chapters. However, note that the differences between the above methods for graph construction are not statistically significant for other measures except the miss rate, as indicated by the p-values at $\alpha < 0.05$ level shown in Table 4.1.

## 4.5 Conclusion

To summarize, in this chapter an association graph was constructed as a model for the contour grouping problem. We first showed the effect of edge detection methods on the quality of grouping, and chose to use Elder & Zucker edge detection. Our goal is to use a multiscale framework in this dissertation, therefore we smooth and downsample images as suggested by [10]. We will start our contour grouping search at scale 3 (quarter resolution) and will then use the hypotheses to find a finer grouping at a higher resolution.

[1] In scale 2, the average miss rate for 30 training images is 5.37% using independent models, and 4.56% using mixture models with diagonal covariances. Also in this scale, the average achievable CMnorm error for 40 validation images is 0.0392 using independent models, and 0.0350 using mixture models with diagonal covariances.

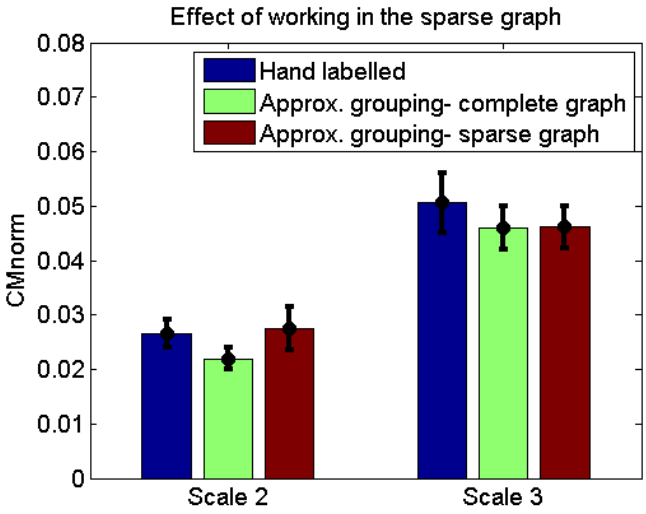



**Figure 4.15: Grouping error introduced by the first stage of forming the association graph**- The average CMnorm error of ground truth hand labellings from HND (Figure 4.7), the shape approximations in complete graphs, and the shape approximations in sparse graphs. Error bars indicate standard error of the mean.

By learning models for pruning the complete graph representation of line segments detected in the image, (a) the search complexity is lowered and (b) an initial grouping inference stage is applied to remove improbable groupings of segment pairs. The cues used for each link in the graph are proximity (log of gap), parallelism, cocircularity, and brightness difference.

Although the differences between independent and Gaussian mixture models in most selection criteria for graph construction methods are not statistically significant, the mixture models with diagonal covariances result in significantly lower miss rates and are therefore used in the subsequent chapters. The error introduced by pruning is relatively small (Figure 4.15).

# 5

# Closed Contour Grouping

The goal of Closed Contour Grouping is to extract a *closed contour* bounding the most *salient* object in the image. The contour grouping problem can be viewed as graph search where nodes model line segments, and links model possible groupings of two line segments. Object boundaries in the image correspond to graph cycles, or elementary circuits; i.e. paths in which the first and last nodes are identical, and no other node appears twice. Furthermore, object boundaries are simple (non-self-intersecting). We will refer to graph cycles corresponding to non-self-intersecting image contours as *simple* closed paths or simple cycles.

Given the sparse association graphs formed in the previous chapter, the goal is to find contour hypotheses for object boundaries in the image. In section 5.1, I will describe a greedy algorithm for extracting simple closed contours. Local cues have been shown to be insufficient for achieving high performance by contour grouping [10]. In section 5.2, I will therefore propose a method for integrating local and global cues in graph path costs. However, depending merely on this path cost to steer the contour extraction algorithm will result in similar paths and low diversity. To overcome this issue, in section 5.3 I will propose a method for promoting diversity in the contour extraction algorithm based on Principal Component Analysis (PCA)[76]. In section 5.4

I will evaluate each of these contributions. Parts of this chapter have been published in [70].

## 5.1 Extracting Closed Contours

Object boundaries in the image correspond to simple cycles in the sparse association graph. The goal is to *find a cycle in the association graph corresponding to a simple closed contour in the image best approximating the boundary of a salient object.*

If these optimal contours could be modelled as first-order Markov, allowing for the decomposition of the grouping cost into local components, then this would be a shortest-path problem that can be solved with Dijkistra's algorithm or dynamic programming [37]. However the Markov assumption cannot be strictly correct, since the constraints of closure and simplicity induce global dependencies. Also, the Markov assumption induces an exponential prior on contour length, leading to a bias toward small bounding contours [9]. Finally, it has been shown empirically that the frequency of high-curvature events is not consistent with the Markov assumption [35].

These deviations from the Markov assumption render exact polynomial methods such as Dijkistra's algorithm ineffective, since they cannot incorporate constraints like self-intersections. For example, the simplicity constraint breaks the required optimal substructure property of the shortest path problem: subpaths of simple shortest paths are not necessarily the shortest simple subpaths.

By using a normalized objective function and taking a discrete optimization approach, the Ratio Contour (RC) approach [17] avoids the bias to small contours and reduces the problem posed by self-intersections. Nevertheless, the RC method does not guarantee simple contours, and the ratio formulation makes it difficult to develop a full probabilistic theory for which optimal parameters can easily be learned.

Note that enumerating all elementary circuits is not a feasible option, since the fastest algorithms [77] have a complexity of $\mathrm{O}\big((n+e)(c+1)\big)$, where there are $n$ nodes,

$e$ edges and $c$ elementary circuits in the graph, resulting in an exponential complexity for large connected graphs.[1]

In pursuing a probabilistic approach, it is important to note [9, 10] that deviations from the strict Markov assumption do not mean that the naive Bayes model (factoring of the likelihood over edges in the cycle) cannot be used as a useful basis for approximate search algorithms that can i) detect and avoid self-intersections, ii) dynamically incorporate global cues and iii) yield multiple candidate closed contour solutions that can then be evaluated using more accurate probabilistic models. In prior work, this approach has proven successful when domain-specific priors are available [9], and also in the context of a multi-scale coarse-to-fine approach, where priors from coarse scales can be fed down into finer scales to narrow the approximate search [10].

In this chapter, we explore an approximate search approach without multi-scale or domain-specific prior. Our algorithm is based upon a greedy constructive search of [9] as explained in 2.2.1.2, replicated here for convenience:

**Algorithm: Closed Contour Extraction**

**Input:** Sparse association graph

**Output:** A set $C$ of cycles in the graph corresponding to candidate simple closed curves.

1- Initialize $C = \{\}$; $m = 1$; $S = \{s_1 = (t_1), s_2 = (t_2), ..., s_N = (t_N)\}$ as set of all paths of length 1 (all nodes).

2- Loop

a. Extend each path $s_i$ in $S$ by one node. Set $m = m + 1$.
b. Discard all paths corresponding to non-simple (self-intersecting) curves.
c. Add all closed paths to set $C$, and remove them from $S$.
d. Calculate the cost of each path in $S$: $W(S) = \{w(s_1), ..., w(s_{n_m})\}$ (see section 5.2).
e. Discard from $S$ all but the lowest-cost $N_m = \frac{N_{mem}}{m}$ paths maintaining diversity (see section 5.3).

Until $m = M$ ( a maximum length for paths)

3- Calculate predicted error for all cycles in $C$, and return the best (see chapter 6).

[1] The number of elementary circuits in a dense directed graph can grow faster with $n$ than the exponential $2^n$ [77, 78].

Note that the above algorithm is an all-source breadth-first search (BFS), informatively pruned at every length to lower time and memory complexity. The maximum curve length $M$ is determined from the ground truth cycles in the training data, and the number of paths $N_m$ to retain at each stage is determined by a specified budget.[1]

Note that our algorithm differs from [9] in (i) calculation of costs for the paths in step (d), (ii) applying diversity in step (e), and (iii) choosing the best cycles in step (3), as will be discussed in the next chapter. In the following section, I will address the key question of how the cost of each path is determined.

## 5.2 Integrating Local and Global Cues in Path Costs

The local cues mentioned in section 4.3 are not sufficient for comparing a set of groupings. Due to their locality, they often "miss the forest for the trees" [10] and get stuck in detail, texture, and clutter in the image. In prior work this problem has been addressed by incorporating top-down priors [10], or mid-level and global cues such as symmetry of parts [21, 79], convexity [22], compactness and color homogeneity [27]. These cues are typically translated into local components in a heuristic fashion, or used only to rank closed contours, once candidate contours have been found.

Here we explore how global cues could be combined with local cues in a more rigorous way, and during the critical phase when closed contour hypotheses are being formed. We develop and test this method using a global colour contrast cue.

The models learned in this section are learned across distinct objects in the training images. To select distinct objects, objects in the SOD dataset are first sorted in decreasing order of consistency among human subjects (MMC), as defined in Equation 3.13. Objects are pruned if their intersection over union with objects preceding in the

[1]The maximum curve length $M$ is set to 60 and 100 for coarse and fine scale (scale =3 and scale=2) respectively. The budget set to limit running time is 4000 lines. Therefore at length $m$, $N_m = 4000/m$ distinct paths are maintained.

sorted list is greater than 50%. The number of distinct objects found in our training dataset of 30 images is 49, with an average of 1.63 per image.

### 5.2.1 Local cues

Deviations from the strict Markov assumption do not mean that the naive Bayes model cannot be used as a useful basis for approximate search algorithms. Assuming conditional independence between local cues for edges in a path $c_k$, the log likelihood ratio for the path based on the set of these local cues is proportional to the log likelihood ratio over its constituent edges. Here we employ the average log likelihood ratio:

$$f_l(c_k) = \frac{1}{m} \sum_{(i,j) \in c_k} \log L_{ij}, \text{ where } m \text{ is the number of links in } c_k \tag{5.1}$$

where $L_{ij}$ is defined in Equation 4.11. The reason for using an average log likelihood (geometric mean) is to obtain a scale invariant local cue used for predicting the path error, as will be explained shortly.

### 5.2.2 Global cue

We define the global colour contrast of path $c_k$ as the symmetric $\chi^2$ distance [27] between normalized colour histograms $H_l$ and $H_r$ of the pixels within two adjacent bands of width $w_c$ on either side of the path, as illustrated in Figure 5.1.

$$f_g(c_k) = \chi^2(H_l, H_r) = \frac{1}{2} \sum_{i=1}^{n_b} \frac{(H_l(i) - H_r(i))^2}{H_l(i) + H_r(i)} \tag{5.2}$$

where $n_b$ denotes the number of bins used for the colour histograms. We explored a number of colour spaces and histogram resolutions, evaluating each by computing a signal-to-noise ratio for $f_g(c_k)$, as a discriminant between paths on ground truth cycles and paths computed using only local cues. We found that averaging the above $\chi^2$ measure over only the L* and a* channels of the Lab color space, using a relatively coarse sampling of 8 bins per channel and a small width of $w_c = 4$ pixels yielded

optimal results.[1] Note that this color cue is different from an averaged local cue, as will be discussed in section 5.4.

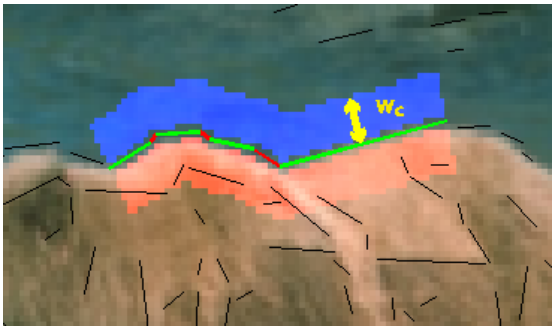



**Figure 5.1: Figure/ground bands used to compute global color contrast cue**

### 5.2.3 Prediction of path errors

The key question is how to combine this global cue with the local cues of proximity, good continuation and brightness contrast. Modeling local grouping decisions using maximum likelihood methods is natural, as there is a clear division between the ON and OFF classes. However, combining these cumulative local cues probabilistically with global cues is a non-trivial problem, partly because the effects of deviations from the naive Bayes assumption accumulate, making the absolute value of the average log likelihood ratio (Eqn. 5.1) unreliable. In addition, at the stage of evaluating paths, the division between ON and OFF is no longer clear: paths can be partially correct. Therefore it is more appropriate at the path stage to adopt a regression approach: given a measure of distance between two paths, learn a regressor that will predict the distance of a candidate path from ground truth based on a number of (both local and global) predictive cues.

As discussed in section 3.2 a natural measure for measuring the distance of a closed path and a reference ground truth contour is the Contour Mapping (CM) measure

[1] The optimal parameter values found for the fine scale (scale 2) are 12 bins and $w_c = 8$ pixels.

(3.2.4). Unfortunately, the CM measure is too computationally expensive to use at this stage of processing, due to the large number of paths that must be evaluated. For this reason, we employ a simpler measure at this stage of processing. In order to make learning efficient, in [70] we used the average distance of the pixels on the path from the closest ground truth contour as a relatively inexpensive measure of the distance of candidate paths from ground truth. Letting $\{p_1^k, ..., p_n^k\}$ represent the points on path $c_k$, the average distance for path $c_k$ from ground truth contour $G$ is:

$$D_G(c_k) = \frac{1}{n} \sum_{i=1}^{n} \min_{p_j \in G} d(p_i^k, p_j) \tag{5.3}$$

This measure evaluates the path without considering how well the ground truth contour is explained, and clutter close to the ground truth contour can get a low error value. To address this, we can also consider the average distance of the ground truth pixels from the path:

$$D_{c_k}(G) = \frac{1}{m} \sum_{j=1}^{m} \min_{p_i^k \in c_k} d(p_j, p_i^k) \tag{5.4}$$

where $m$ is the number of pixels on the ground truth contour G.

The average of the above distances is the mean distance (MD) measure [57] discussed in section 3.2.2. However, since the fragments are open contours, I will use the following modified mean distance considering only a subset of at most $n$ ground truth pixels closest to the path $c_k$:

$$\epsilon(c_k, G) = \frac{1}{2} \left( D_G(c_k) + D_{c_k}(G') \right) \tag{5.5}$$

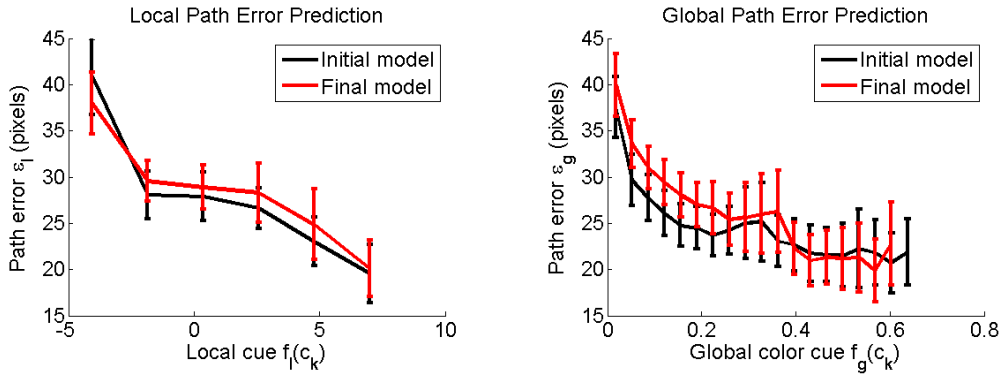
where

$$G' \subseteq G, |G'| = \min(m, n), \text{ and}$$

$$\min_{p_i^k \in c_k} d(p'_j, p_i^k) \leq \min_{p_i^k \in c_k} d(p_j, p_i^k), \forall p'_j \in G' \text{ and } \forall p_j \in G \setminus G'$$

The tricky part of this regression problem is to acquire appropriate training data: we need plausible candidate paths in order to learn appropriate weights for combining cues, however we also need to use these cues in order to determine plausible paths. This chicken and egg problem was solved by training the models in two iterations. In

the first iteration, we run the contour extraction algorithm, setting the cost of paths to the negative average log likelihood ratio, $-f_l(c_k)$. We use the paths obtained in this iteration to learn an initial regression model for combining cues. This initial model is used in a second run of the contour extraction algorithm, using both local and global cues in determining the cost of paths. We use the paths obtained in this second run to learn the final regression model (see Figure 5.2).




(a) (b)

**Figure 5.2: Learned nonparametric predictors for path error** based upon (a) local and (b) global cues. Error bars indicate standard error of the mean among models learned for distinct objects.

We take a nonparametric approach to the regression problem, binning the local cue $f_l(c_k)$ and global cue $f_g(c_k)$, and then estimating the mean $\hat{\epsilon}_l$ and $\hat{\epsilon}_g$ and variance $\sigma_l^2$ and $\sigma_g^2$ of predicted error for each. Assuming independence between the local and global cues, and normal distribution of error values at each cue bin, the least-squares error prediction $\epsilon_{ML}$ (maximum likelihood for normal distributions) is then given by [80]:

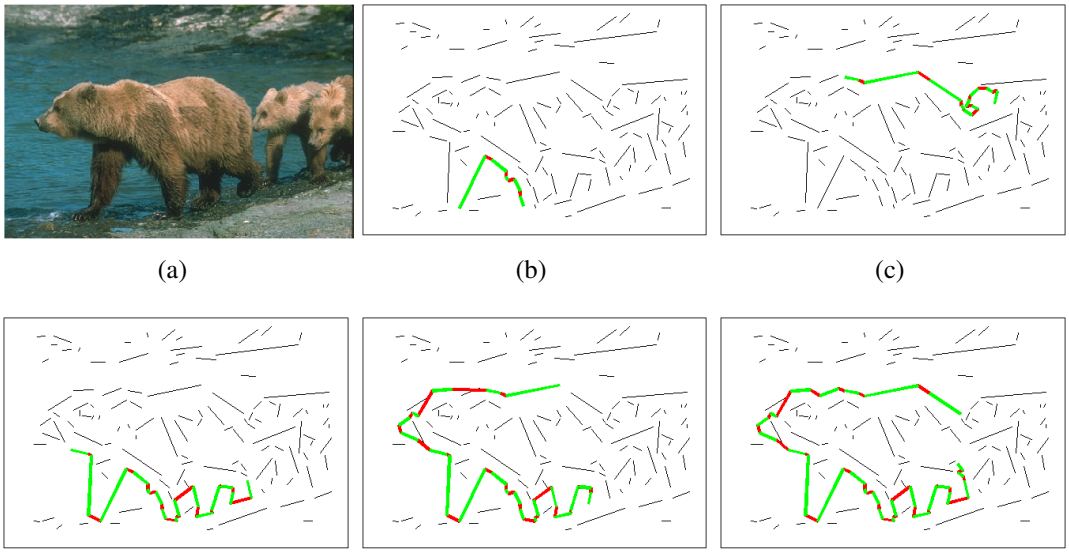
$$\hat{\epsilon}_{ML} = \frac{\hat{\epsilon}_l/\sigma_l^2 + \hat{\epsilon}_g/\sigma_g^2}{\sigma_l^{-2} + \sigma_g^{-2}} \tag{5.6}$$

This predicted error is used as the path cost in Step 4 of our closed contour extraction algorithm (Section 5.1).[1]

[1]The regression model is an average across nonparametric models obtained for each distinct object.

Note that most of the training paths are paths with high error. Using parametric models would bias the parameters towards this majority, leading to almost no learning for the minority of samples with low path errors. The nonparametric model used above, however, is less biased by the majority since the model is more local and can learn the behaviour of low error paths falling within same bins of the nonparametric model.

Figure 5.3 shows the path with the lowest cost at different iterations of the contour extraction algorithm for a sample image.



(a) (b) (c)

(d) (e) (f)

**Figure 5.3: Samples of paths with lowest cost in contour extraction algorithm**. (a) original image, and lowest cost paths at(b) length 5, (c) length 10, (d) length 15, (e) length 20, and (f) length 25.

Only paths shorter than the best closed contour for each distinct object were used to learn the nonparametric models, where the best closed contour is defined as the contour with lowest modified mean error $\epsilon(c_k, G)$.

## 5.3 Promoting Path Diversity

Due to the exponential size of the search space, as paths grow there tend to be many highly similar paths with low predicted error. This can have a negative effect on performance because distinctive paths that look slightly less promising at an early stage may be pruned out early, before their strengths are recognized. To address this problem, at each iteration of the closed contour extraction algorithm we cluster the candidate paths and then select representatives from each cluster. To make clustering efficient, we represent each path $c_k$ by an indicator vector $\mathbf{v}_k$ that identifies the segments on the path (but not their order), and then do Principal Component Analysis (PCA) on these indicator vectors, weighted by the inverse of their predicted errors. By sequentially selecting paths with maximum projection on the leading eigenvectors, we promote paths that are the leading representatives of low-error clusters, while ensuring that all leading clusters are represented.

**Algorithm: Path Diversity**

**Inputs:** A large set of candidate paths $S = \{c_k\}$ and their predicted errors $E = \{\hat{\epsilon}(c_k)\}$
**Output:** A small set of $N_m$ selected paths $S' = \{c'_k\}$

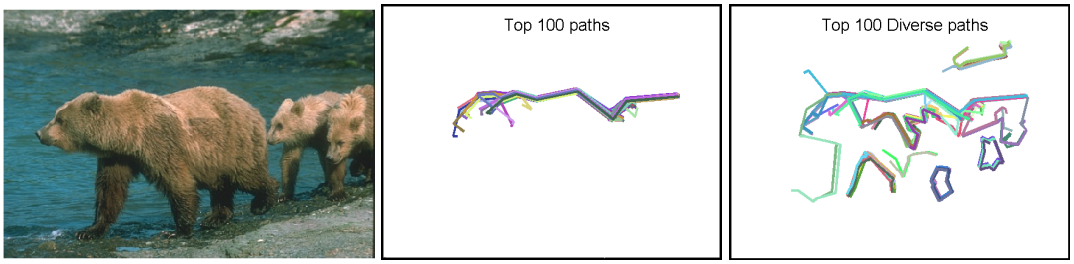
1. Initialize $S' = \{\}$
2. Represent each path $c_k \in S$ as an $N$-vector $\mathbf{v}_k$, where $N$ is the number of segments in the image. If segment $i \in c_k$, set $\mathbf{v}_k(i) = \frac{1}{\hat{\epsilon}_k}$, otherwise $\mathbf{v}_k(i) = 0$.
3. Compute the principal components $U = \{\mathbf{u}_k\}$ of $V = \{\mathbf{v}_k\}$.
4. Until $\|S'\| = N_m$:

   For $k = 1$ to $N$

   i. Find fragment $c_k \in S$ with the largest projection on $\mathbf{u}_k$.
   ii. Add $c_k$ to $S'$ and remove from $S$.

Figure 5.4(b) shows the top 100 lowest cost paths of length 5 all concentrated on the salient boundary on the back of the bear. Figure 5.4(c) shows the top 100 paths

after application of our diversity algorithm.




(a) (b) (c)

**Figure 5.4: Top 100 paths with and without PCA diversity**. (a) original image, (b) top 100 lowest cost paths of length 5 (before applying diversity), and (c) top 100 diverse paths of length 5 (after applying our path diversity algorithm).

## 5.4 Evaluation of Contour Extraction Method

### 5.4.1 Effect of the Global Colour Cue

The global colour contrast cue suggested above is quite distinct from a local colour contrast cue computed at each segment. This cue selects for paths where there is a coherent and consistent difference in the colours found in the figure from the colours found in the background. In the local cue, this coherence is lost: a path with identical global figure/ground colour distributions could be selected as long as colour contrast is locally high. Local colour contrast is measured locally and then converted to a log likelihood ratio given ON and OFF learned models, that is then summed over the path, similar to other local cues.

Figure 5.5 shows a comparison of the effect of global versus local colour contrast on the minimum error of closed contour hypotheses averaged over 40 validation images. The minimum error achieved over the 40 validation images shows on average that the global colour is slightly better than using no colour information, and is as good as using local colour. Pairwise repeated measure t-tests show that the effect of

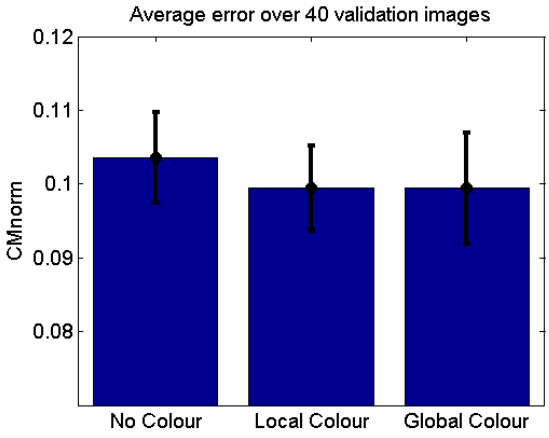



**Figure 5.5: Effect of global colour contrast cue** - Comparing minimum error among contour hypotheses obtained in 40 validation images, on average using global colour contrast is only slightly better than using no colour, and as good as using local colour. Error bars indicate standard error of the mean.

| | No Colour | Local Colour |
|---|---|---|
| Global Colour | 0.4450 | 0.9936 |

**Table 5.1: p-values for pairwise repeated measures t-tests showing the effect of global colour contrast cue**- The effect of the global colour contrast cue on the quality of the best closed contour produced by the algorithm is not statistically significant at the $\alpha \leq 0.05$ level.

the global colour contrast cue on the quality of the best closed contour produced by the algorithm is not statistically significant (see Table 5.1). This result might be due to insufficient data.

### 5.4.2 Effect of Diversity

To show the effect of the path diversity method proposed in the previous section, we compared the minimum CMnorm error of closed contour hypotheses against SOD ground truth averaged over 40 validation images, with and without diversity (Figure 5.6). In the *No Diversity* case, all but the lowest-cost $N_m$ paths with minimum estimated cost are discarded in each iteration of the contour extraction algorithm. The

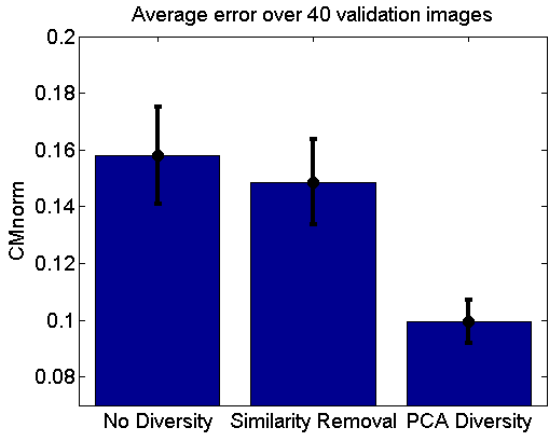



**Figure 5.6: Effect of maintaining path diversity** - The proposed diversity method, PCA diversity, on average results in a lower minimum error among contour hypotheses obtained in 40 validation images when compared with applying no diversity at all, or applying a similarity removal method. Error bars indicate standard error of the mean.

| | No Diversity | Similarity Removal |
|---|---|---|
| PCA Diversity | **5.78e-4** | **6.05e-4** |

**Table 5.2: p-values for pairwise repeated measures t-tests showing the effect of PCA diversity**- The effect of the PCA diversity on the quality of the best closed contour produced by the algorithm is statistically significant at the $\alpha \leq 0.05$ level.

PCA method is proven to be much more effective in the contour extraction algorithm.

A simpler method for maintaining diversity is to remove all but the lowest cost path passing through the same set of line segments (but in a different order or orientation), prior to choosing the top $N_m$ paths. This method is used in the implementation of [10]. However, as shown in Figure 5.6, this *Similarity Removal* method is not as effective as our proposed PCA method. The effect of the PCA diversity on the quality of the best closed contour produced by the algorithm is statistically significant at the $\alpha \leq 0.05$ level (see Table 5.2).

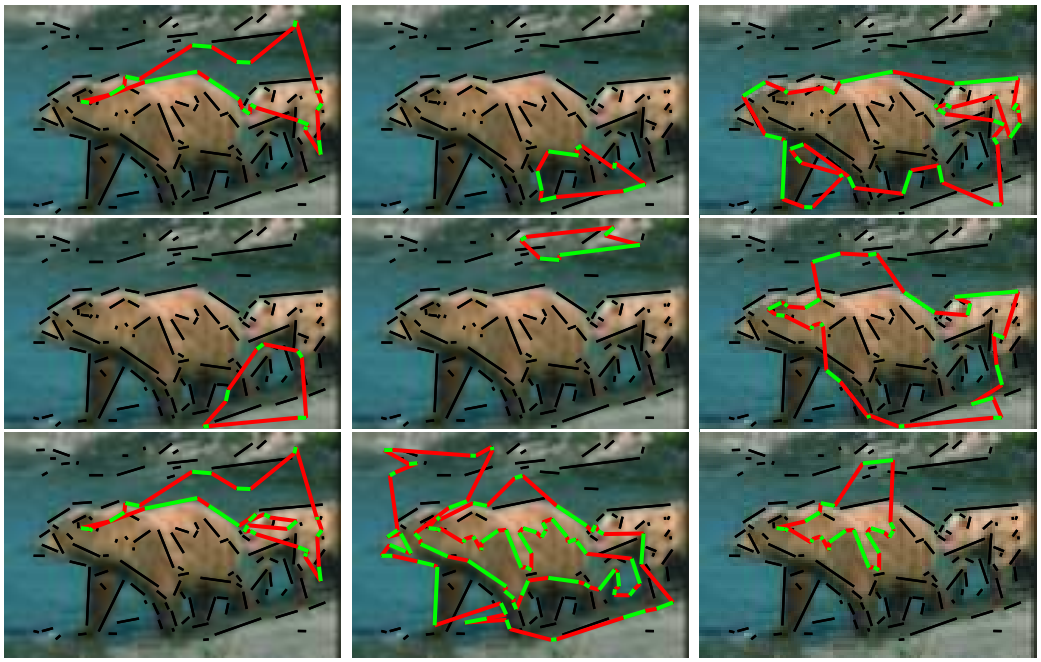


**Figure 5.7: Examples of closed contour hypotheses extracted in a sample image.** Samples are selected randomly from the set of closed contour produced for this image.

## 5.5 Conclusion

In this chapter, an effective and novel method for combining local and global cues was introduced at the stage of forming new closed contour hypotheses. However, comparisons with using no colour or local colour show that the global colour contrast cue does not have a statistically significant effect on the quality of the best closed contour produced. This might be due to insufficient data.

In addition, a novel method for promoting diversity in the formation of contour hypotheses is suggested which leads to substantial improvements in performance. The result of this stage is a set of closed hypotheses including perceptually plausible closed contours.

Figure 5.7 shows randomly selected samples of closed contour hypotheses produced for a sample image. In the next chapter, we will discuss the ranking of these contours and a method for selecting the best of these for output.

# 6

# Ranking of Closed Contours

The greedy search produces a set of closed contour hypotheses based on both saliency and diversity. These contours need to be ranked for output. Once closed contours have been computed, there is opportunity for refining our error predictions. Since each closed contour partitions the image into "inside" and "outside" regions, new contour features can now be computed. In addition, due to the voting from the set of closed contours produced in the image, we can make some inference about foreground and background in the image which can be used in the ranking of the contours, as will be discussed shortly.

In the contour grouping literature, the grouping cost of Wang et al. [17] is very popular. This grouping cost is defined as the ratio of the grouping cost along the contour (e.g. total gap) to the regional information enclosed by the contour (e.g. total area, or homogeneity of enclosed area). The total gap over area (GoA) cost function penalizes weakness in local cues along the contour, while rewarding larger enclosed regions. Since this ratio is correlated with the perimeter to area ratio, it also implicitly rewards compactness and circularity. This ratio has often been used for measuring the quality of contour hypotheses [17, 79].

Estrada and Jepson [27] argue that the quality of the contour is related to the quality

of the segmentation induced by that contour and measure this quality by estimating the encoding cost as the sum of the log-probabilities of drawing each pixel from the labelled regions given the colour histograms.

To measure the quality of the closed contours, we introduce a more general method that is able to include the above saliency features, as well as other cues. We repeat the approach we took to integrating local and global cues in path costs (Section 5.2), regressing error on a set of closed contour features. This estimated error will be used to rank and select the best contours, as will be explained in section 6.1.

In addition to the estimation of error, and similar to the results seen in the constructive phase, we need a method for choosing diverse contours to improve performance. The proposed method for maintaining diversity is explained in section 6.2. Parts of this chapter have been published in [70].

## 6.1 Prediction of Error for Closed Contours

Similar to the previous chapter, a nonparametric regression of the predicted error given contour features is used as a model to predict the quality of closed contour hypotheses. Here, however, we use the log[1] of the CM measure [50] explained in section 3.2.4, normalized by the square root of the area of the ground truth contour, as a measure of error. As shown previously, the CM error measure corresponds closely with human perception. Since far fewer closed contour candidates are computed than open path candidates, the computational cost at learning time is manageable.

As predictors, we experimented with features representing: (i) the size, (ii) the complexity, (iii) the strength of local cues, and (iv) the colour saliency of the contour. We tried different ways of measuring these cues and chose the combination that resulted in the best performance in a validation set of 40 images. In the following sec-

[1] Using log of the CMnorm error was found to improve the normal approximation of the distribution of error, required for the cue integration method (Equation 6.4).

tions, I will show the effect of each of these cues on the ranking of the contours, while using the best performing measure for the other three cues.

Recall that for each of the validation images, there are a number of salient ground truth object boundaries generated by multiple human subjects in the SOD dataset. The error of an algorithm-generated contour is defined as the minimum over all ground truth contours for that image. I will use the normalized CM measure as the error measure. To evaluate object segmentation algorithms as hypothesis generators, it is common practice [20] to allow the algorithm to generate multiple ($n$) hypotheses evaluated according to the **minimum** error over these $n$ hypotheses. Varying $n$ then sweeps out a performance curve (e.g. Figure 6.1).

### 6.1.1 Ranking by size

The constructive phase is a greedy search that starts from short paths and proceeds to longer paths, saving closed contours formed in this process. Shorter contours have a higher chance of being formed, but many of them correspond to non-salient objects or simply clutter. Therefore, as we would expect, the predicted error is lower for longer contours relative to shorter ones. Figure 6.1(a) shows the effect of the size cue on ranking of validation images. Using the log of the perimeter of the contour as a ranking cue results in lower errors among the $n$ ranked contours than using the log of the contour's area, and lower than using both. Although for our validation set, using no size cue seems to be just as good as using the perimeter cue, the size cue was found to be an important cue in the training set. This inconsistency might be due to insufficient data and the fact that the ground truth objects in the validation set are on average smaller than those in the training set, as shown in Figure 6.2. We decided to use the log of perimeter as a size cue in our ranking model.

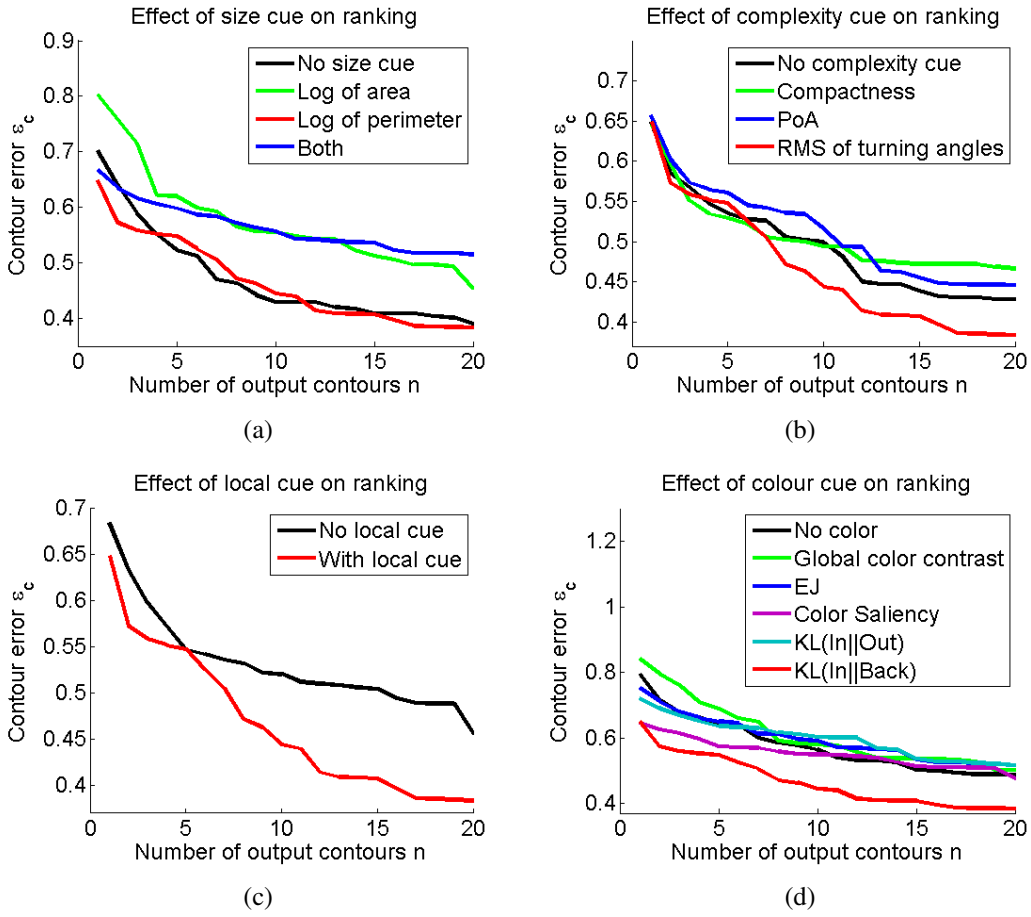
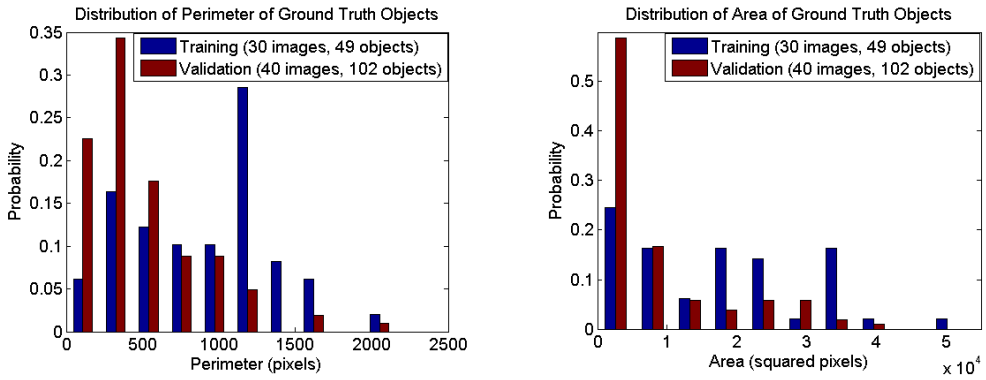



**Figure 6.1: Effect of cues on ranking performance in validation set** (a) size cue, (b) complexity cue, (c) local cue, and (d) colour cue.

### 6.1.2 Ranking by contour complexity

Contours with low errors are usually not very short; however, very long, wiggly, or complex contours are also not representative of salient object boundaries. Salient objects often have smooth, compact, and convex boundaries, with few concavities due to parts. Grouping method in the literature typically control the complexity of the contours. For example, grouping methods minimizing the total gap over area (GoA) ratio ([11, 17, 19, 20, 21, 22, 79]) promote circularity and compactness since this ratio is correlated with the *perimeter over area ratio (PoA)*. In the grouping method of Estrada




(a) (b)

**Figure 6.2: Distribution of size of ground truth objects**- The distinct ground truth objects in the training set are on average larger than those in the validation set, as shown by (a) distribution of object perimeters, and (b) distribution of object areas.

and Jepson [27, 81] contours with low *compactness* values, defined as the ratio of the area of the contour to the area of its convex hull, are discarded. In the multiscale algorithm [10] low complexity is promoted by spatial priors obtained from low resolution images. The number of line segments is often low in the low resolution images and therefore spatial priors are often of low complexity.

In our ranking model, we used the root-mean-square (RMS) of *turning angles* as a complexity feature. To attenuate noise introduced, the sequence of turning angles was first smoothed with a Gaussian[1]. Figure 6.1(b) shows the effect of this feature and compares it to other possible complexity features.

### 6.1.3 Ranking by local cue

Using the local cue (defined in 5.1) as a ranking cue is useful as can be seen in Figure 6.1(c).

[1]The variance of the Gaussian used is 3.

### 6.1.4 Ranking by colour

Colour is an important cue for human vision. As we showed in the previous chapter, using a *global colour contrast cue* defined as $\chi^2$ difference between colour histogram of the two sides of a path (5.2) positively affects the grouping performance. Colour is also a very strong cue in ranking closed contours.

Estrada and Jepson [27] rank their contours using an encoding cost based only on colour histograms. Their encoding cost is defined as the sum of log-probabilities of drawing each pixel in the image from the appropriate region:

$$\text{EJ} = -\sum_{x \in \text{In}} \log p(x|H_{\text{in}}) - \sum_{x \in \text{Out}} \log p(x|H_{\text{out}}) \tag{6.1}$$

where $H_{\text{in}}$ and $H_{\text{out}}$ represent the normalized colour histogram of inside and outside regions of a contour, denoted by In and Out, respectively. Assuming the CIELab colour space for the image, the probability of drawing a pixel's colour values $x = (x_L, x_a, x_b)$ from a histogram $H$ is defined by $p(x|H) = H_L(x_L) * H_a(x_a) * H_b(x_b)$ (assuming conditional independence between colour components) where $H_L$, $H_a$ and $H_b$ are the histogram components corresponding to the $L$, $a$, and $b$ colour components, and $H_y(x_y)$ is the value of the histogram corresponding to the bin $x_y$ falls in.

KL divergence is a non-symmetric measure of difference between two probability distributions. More specifically, $KL(P||Q)$ is a measure of information lost when Q is used to approximate P, i.e. the expected number of extra bits required to code samples from P when using a code based on Q. An informative ranking cue is the KL divergence of the normalized colour histogram of the outside region from the normalized colour histogram of the inside region of a contour, defined as:

$$\text{KL}(H_{\text{in}}||H_{\text{out}}) = \sum_i \log\left(\frac{H_{\text{in}}(i)}{H_{\text{out}}(i)}\right) H_{\text{in}}(i) \tag{6.2}$$

where the sum is taken over all bins $i$ in the normalized histograms. When multiple colour channels are available, the average KL value is often used. We will call this cue KL(In||Out) for short.

Achanta et al. [82] suggested a simpler approach using the average colour saliency

value of the pixels inside the contour, where the saliency of each pixel $i$ in the colour saliency map is defined as:

$$S(i) = \sum (L(i) - \bar{L})^2 + (a(i) - \bar{a})^2 + (b(i) - \bar{b})^2 \tag{6.3}$$

where $L(i)$, $a(i)$ and $b(i)$ are the CIELab colour values at pixel $i$ after a Gaussian blurring of the image, and $\bar{L}$, $\bar{a}$, and $\bar{b}$ are the average value for each colour component of this blurred image.

A source of information available in our method is the collective information about foreground and background obtained from all contours in the set of closed contour hypotheses. If each contour votes for the pixels inside it as foreground, a frequency map can be obtained (see Figure 6.3).[1] Based on comparison of normalized frequency values[2] for foreground and background pixels, pixels with frequency values less than 0.1 are assumed to be on the background at time of inference. The KL divergence of normalized colour histogram of this background region from normalized colour histogram of inside region of contours, shown in short as KL(In||Back), was found to have the best ranking performance among the colour cues, as shown in 6.1(d). This cue is therefore used in our ranking method.

### 6.1.5 Prediction of error

Based on results shown in Figure 6.1, we use the following contour features as error predictors:

1. Log of the contour perimeter
2. Root-mean-square of the turning angles on the contour after smoothing
3. Local cues as defined in equation 5.1

[1] All contours have equal votes and they vote the same value for all pixels inside them. An alternative method is to allow a different weight for vote of each contour based on some quality estimate for the contours. This would require two rounds of quality estimation for the contours. The first approach of equal weights for votes of all contours was taken for simplicity.

[2] The frequency map obtained for each image is normalized by the number of closed contours.

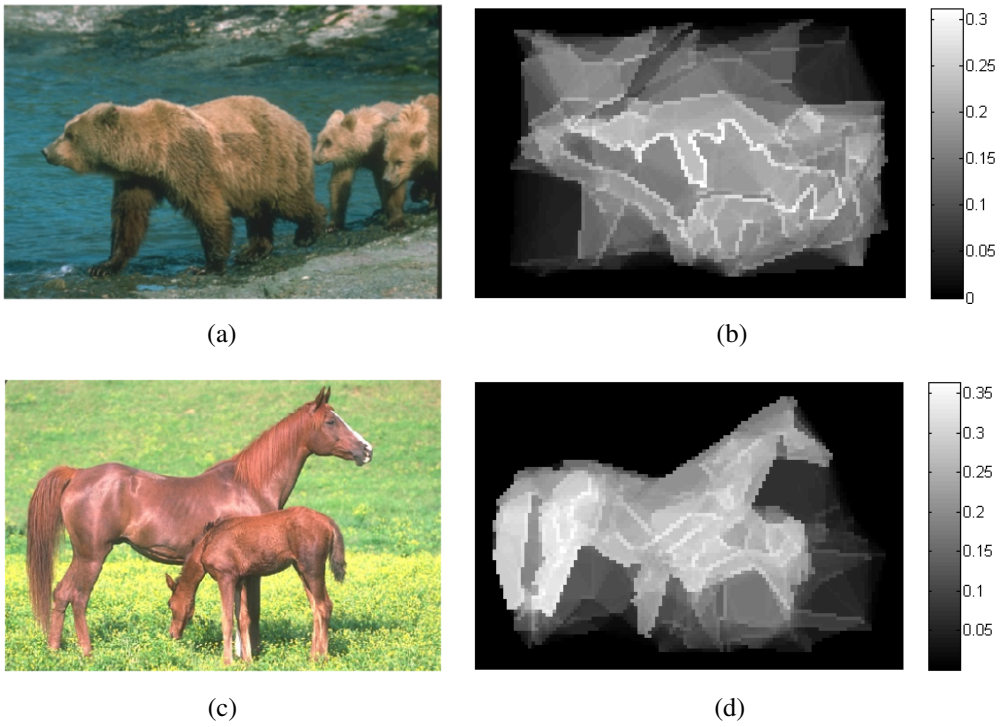



**Figure 6.3: Background detection using frequency maps.** (a) and (c): Sample images from training set, (b) and (d): Frequency maps obtained from the whole set of closed contour hypotheses available for each image. These frequency maps can be used to estimate background colour histograms. (See text for details.)

4. The Kullback-Leibler divergence of colour probability distribution of background region from colour distribution of inside region of the contour

The predicted error is then given by

$$\hat{\epsilon}_{ML} = \frac{1}{\sum_i \sigma_i^{-2}} \sum_i \hat{\epsilon}_i / \sigma_i^2. \tag{6.4}$$

where $\hat{\epsilon}_i$ is the estimated error for cue $i$ and $\sigma_i^2$ is the variance of this estimate [80]. Figure 6.4 shows the nonparametric predictors for contour error of closed contours. As in the previous chapter, this model is the average of models learned for distinct objects in the training set.

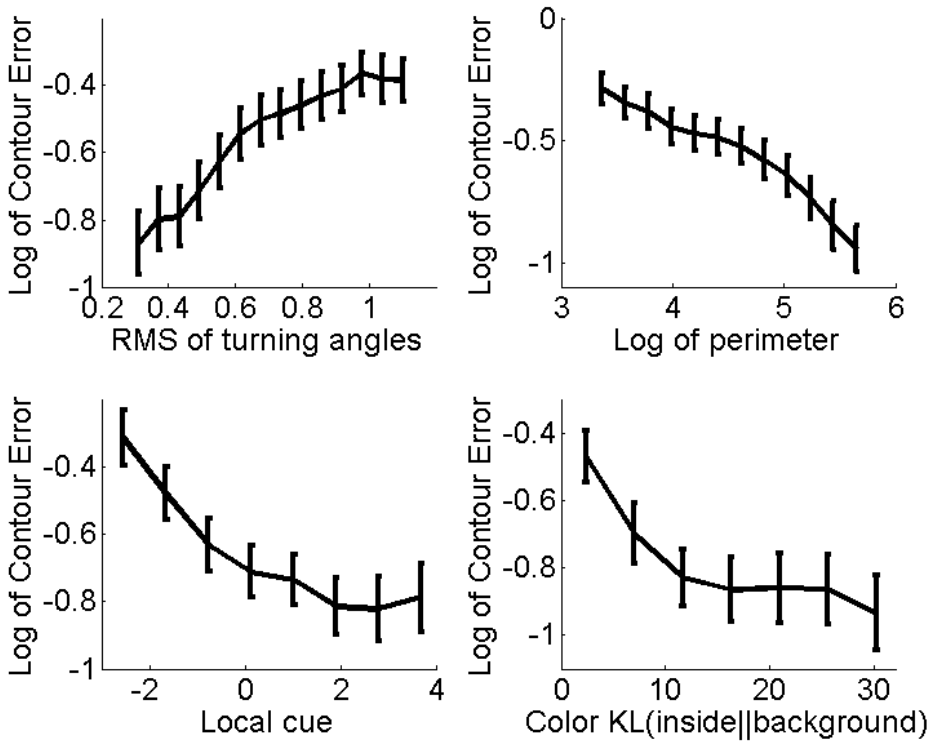



**Figure 6.4: Learned nonparametric error predictors for closed contours**- Please see text for details. Error bars indicate standard error of the mean among models learned for distinct objects.

## 6.2 Removing redundant contours

Just as candidate paths can become overly clustered without promoting diversity, in the ranking of extracted closed contours, often the top-ranked candidates are highly similar approximations to the same object. This is a problem for two reasons:

i. There are often several salient objects in the image, and this clustering may mean that some objects are very far down the list.

ii. In the unhappy event that a false grouping leads to low predicted error, the correct solutions may have very low rankings.

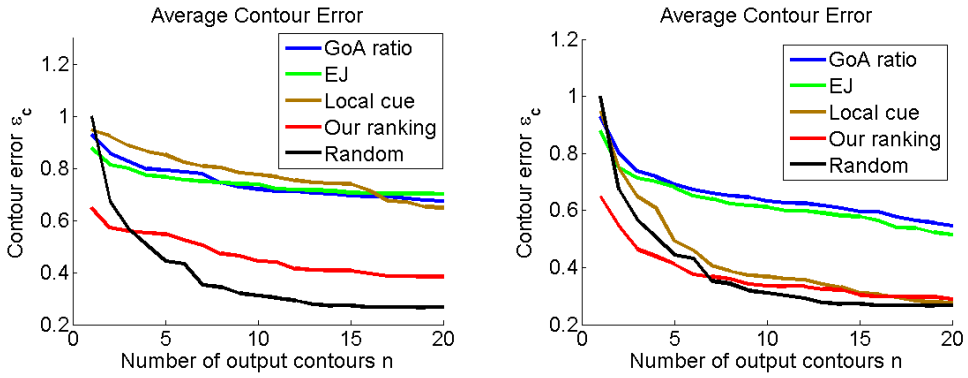



**Figure 6.5: Effect of diversity in ranking contours**- Comparison of our ranking method versus commonly used measures used in the literature and also random selection in the validation set, (a) before applying diversity and (b) after applying a similarity removal diversity method.

Figure 6.5(a) shows a comparison of our ranking method versus other ranking methods reported in the literature and also versus a random selection of contours. The plots show the minimum error over $n$ algorithm contours averaged over 40 validation images, selected by various ranking methods. Our ranking method, as defined in 6.1, outperforms ranking by the total gap over area (GoA) ratio (the ranking measure used in [11, 17, 19, 20, 21, 22, 79]) and also ranking by the colour encoding cost of Estrada and Jepson (6.1) (used in [27]).[1] However, for higher values of $n$ random selection outperforms all!

Figure 6.5(b) shows the effect of using a simple method for removing redundant contours. The method selects closed contours in ranked order, but skips any that share more than a given proportion $\alpha$ of its segments with a contour that has already been selected. Lower values of the parameter $\alpha$ result in higher diversity among selected closed contours. We choose $\alpha = 0.6$ based on performance results on our validation set.

[1]Note that we are applying only the ranking methods used by these authors, and not their grouping methods. The various ranking methods are being applied on the same set of closed contours obtained by our method.

Figure 6.6 - 6.8 show the top 20 ranked closed contours for two sample training images, after removing redundant contours. In addition, the closed contour with the lowest CMnorm error among the set of all[1] closed contours is also shown. This contour is the best output our algorithm could produce, if an oracle ranking method were available. Although most of the contours shown in these figures are not representative of the salient objects in the image, some top contours can capture parts or most of the salient boundaries. As we will show in Chapter 8, our method is still outperforming other available contour grouping methods in most cases.

### 6.2.1 Why not PCA diversity?

In the previous chapter, the PCA technique was used to promote diversity of open contour fragments in the contour formation stage (Section 5.3). Having diverse fragments in the greedy search helps in exploring the search space and therefore results in better contour hypotheses among the set of closed contours output at that stage.

However, the PCA diversity performs poorly in the ranking stage, relative to the above simple method of removing redundant contours, presumably because it selects contours that are too different from each other, at the expense of higher predicted (and actual) error.

## 6.3 Conclusion

In this chapter, the ranking of a set of closed contours and the selection of a diverse subset for output was discussed. A set of predictors describing the size of the contour, its complexity, and the strength of local cues along the boundary were used. In addition, the complement of the whole set of closed contour hypotheses can be used to help estimate a background colour model. The KL divergence of the color probabil-

[1] On average 8,298 closed contours are extracted by our algorithm for each image in the training set.

ity distribution of this estimated background region from the color distribution of the region inside the contour was used as a fourth ranking feature. Nonparametric regression models of these predictors were used for predicting the error of closed contours. The closed contours were then ranked for output based on this estimated error and lower-ranked redundant contours were removed.

Our ranking method outperforms other methods suggested in the literature in the task of ranking a set of closed contours, and it performs better than random for a small ($n < 8$) output size. However, it performs worse than random for $n > 8$. This shows there is potential for improving both the error estimation method and the diversity method.

In the next chapter, we will explore how closed contour hypotheses computed at low resolution (coarse scale) can be used as spatial prior at a finer scale. Working in fine scale could improve the details of the object boundary. Since the search at fine scale is guided by the results obtained at the coarse scale, there is a higher probability of staying on the boundary of the salient object, and not being distracted by clutter.

We will refer to our grouping algorithm presented so far in this thesis as the Enhanced Grouping (EG) algorithm.

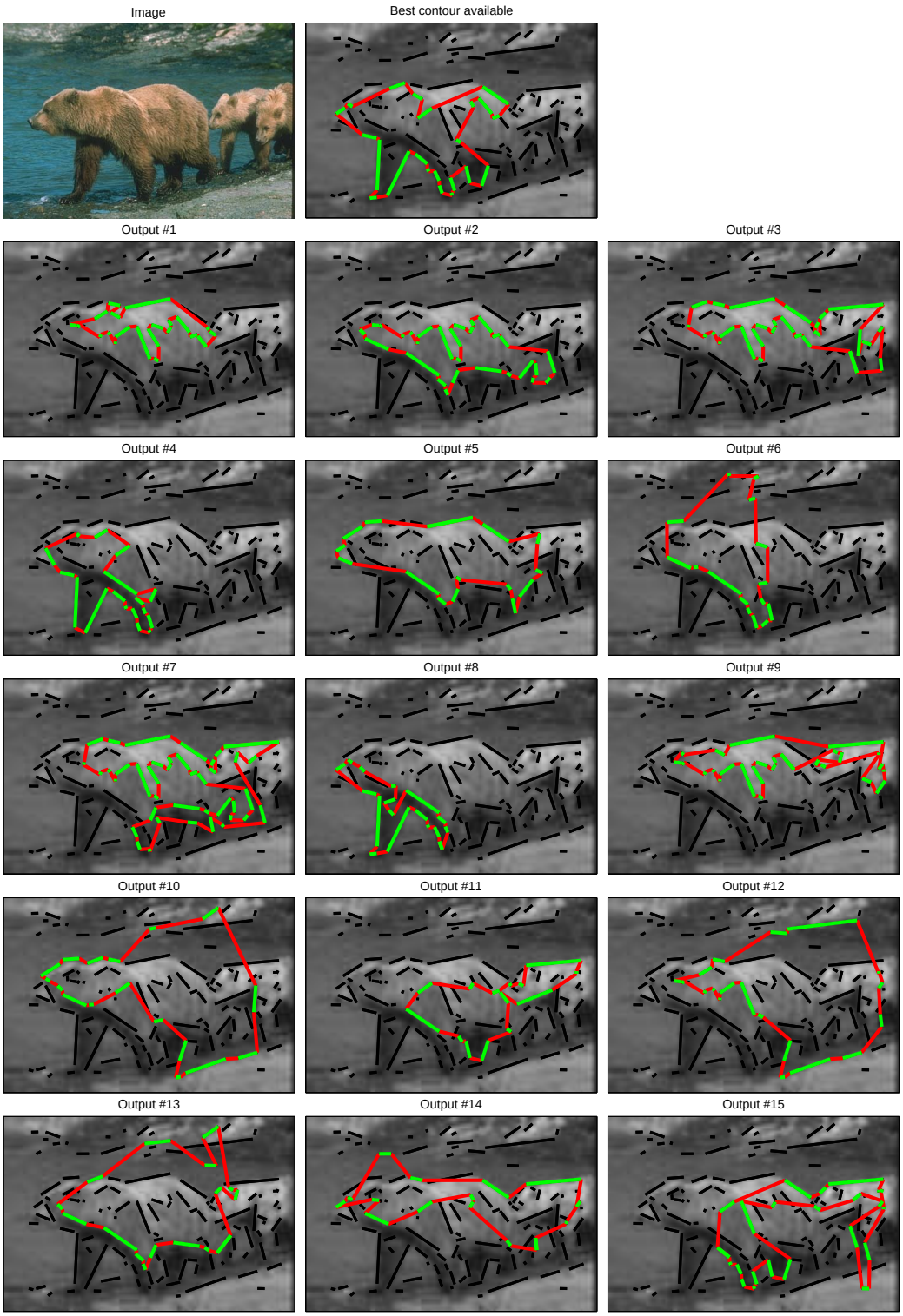



**Figure 6.6: Examples of ranked contours (sample image #1)**- Image and the contour with lowest error available among the set of closed contours, followed by the ranked output contours.

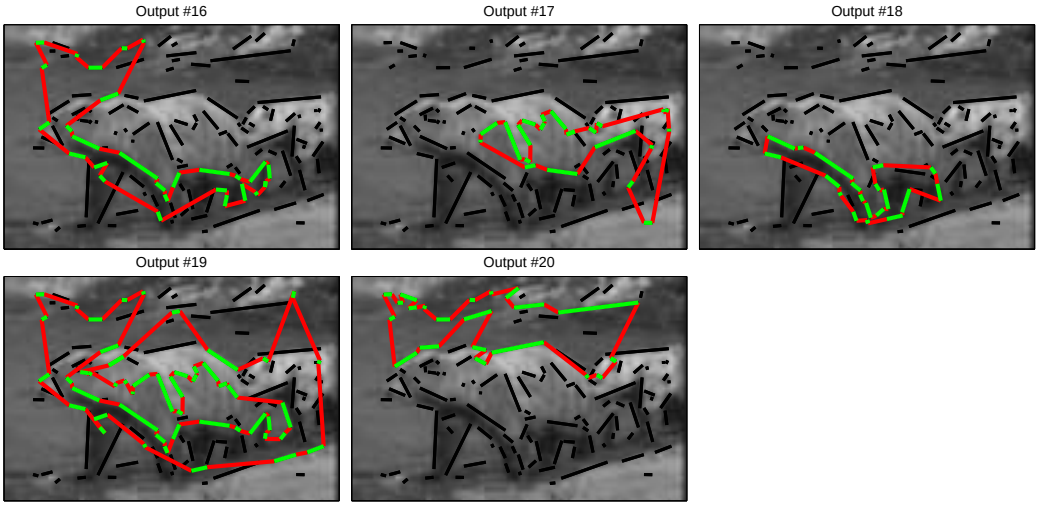



**Figure 6.7: Examples of ranked contours (sample image #1)- continued**.

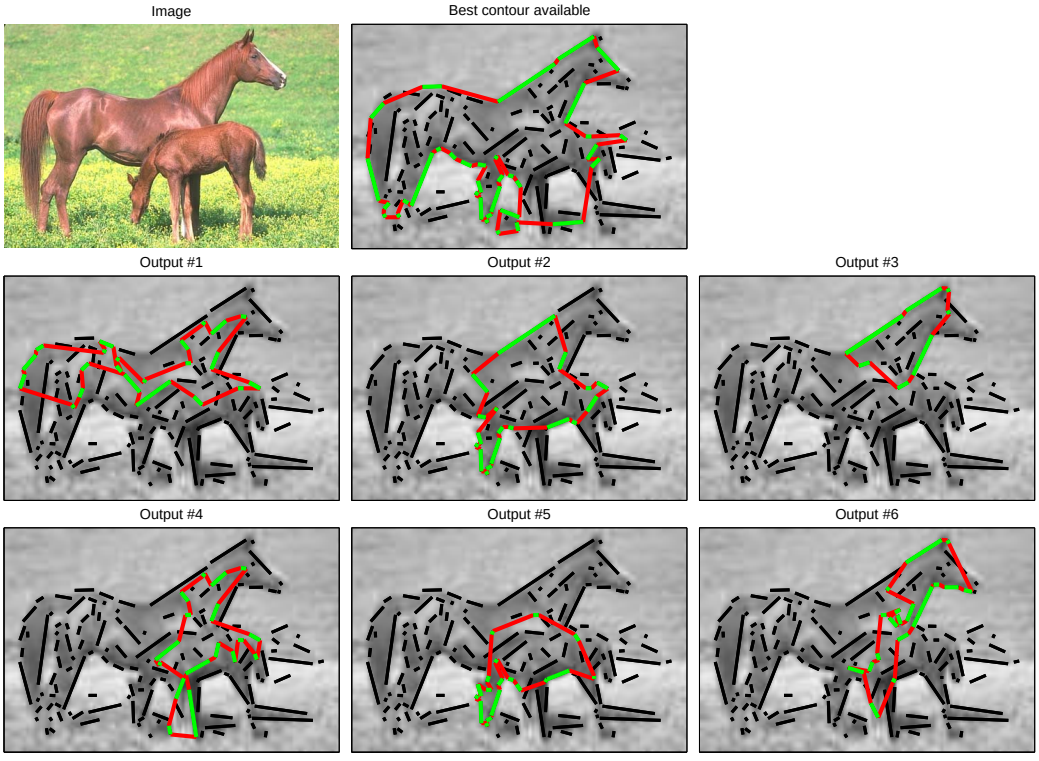



**Figure 6.8: Examples of ranked contours (sample image #2)**- Image and the contour with lowest error available among the set of closed contours, followed by the ranked output contours.

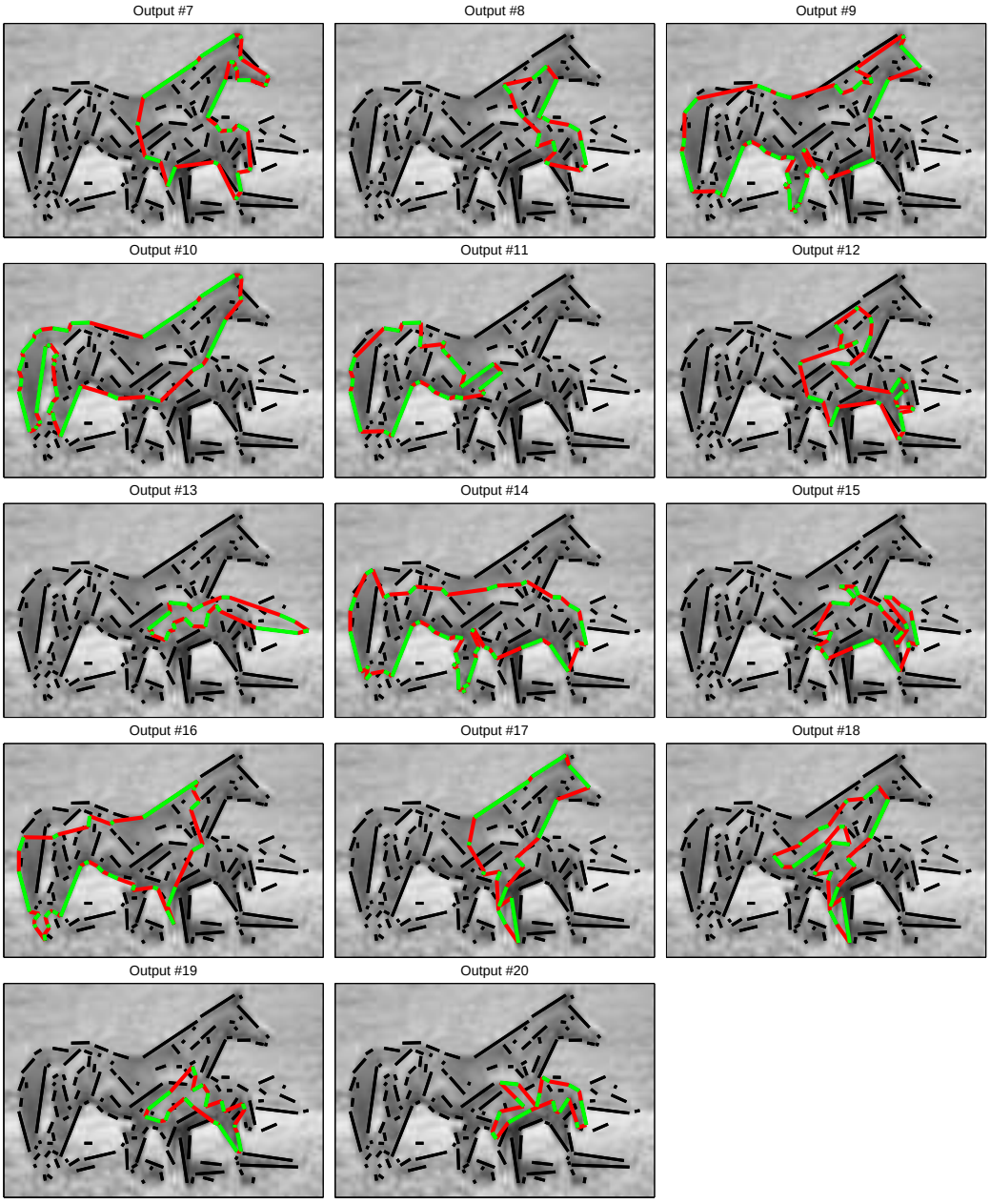



**Figure 6.9: Examples of ranked contours (sample image #2)- continued.**

# 7

# Contour Grouping with Multiscale Prior

It is believed that the human visual system takes advantage of higher level knowledge and top-down processing [83] for object segmentation and detection. This has motivated the use of prior models of objects in combination with other grouping cues in the literature. Elder et al. [9] used approximate polygonal models of boundaries of lakes in their contour grouping method. In addition to using cues specific to lake images, each line segment considered in the grouping is assigned a weight based on its distance to the polygonal lake model and the angle between the line segment and the polygon (two unary cues).

In place of priors based on domain knowledge, the multi-scale approach suggested by Estrada and Elder [10] propagates object contours obtained in a coarse resolution version of the image to a finer scale, where they are used as spatial priors. Search at coarse resolution has the benefit of avoiding clutter and distractions due to detail and texture and thus "missing the forest for the trees" [10]. However, this also means that the resulting object contours lack fine detail. The closed contours obtained at a coarse scale can be used to guide search at fine scale to obtain more detailed boundaries.

Throughout the rest of this dissertation, fine scale refers to scale 2, in which images have half the resolution of the original images in each dimension; and coarse scale refers to scale 3, in which images have a quarter of the resolution of the original image resolution in each dimension. We will start our contour grouping search at scale 3 and will then use the hypotheses to find a finer grouping at a higher resolution[1].

In this chapter, I will investigate using multi-scale spatial priors as suggested by Estrada and Elder [10] in the framework suggested in previous chapters. In the following sections, I will explore using the multiscale priors in

- Extracting closed contours in the constructive phase, and
- Ranking closed contours

## 7.1 Spatial prior in the constructive phase

Three sets of closed contours are explored:

1. Closed contours obtained at coarse scale. We will denote this set of contours as Single-Scale-Coarse (*SS-Coarse*).

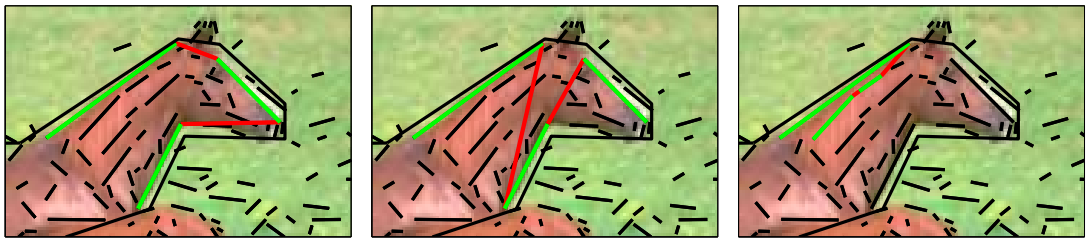
2. Closed contours obtained at fine scale, also by a search guided by local and global cues. We will denote this set of contours as Single-Scale-Fine (*SS-Fine*).

3. Closed contours obtained at fine scale, using a spatial prior obtained from coarse scale. We will denote this set as Multi-Scale-Fine (*MS-Fine*).

Mapping priors from coarse to fine scale requires upsampling. This can be done using a truncated Fourier descriptor representation [10] or by simply up-scaling the polygon. We used the second option for simplicity[2]. The top $K$ ranked contours

[1] As shown previously in Figure 4.6, there is a greater impact from going from scale 2 to 3 than going from scale 1 to 2, both in the number of lines and the error introduced.

[2] Preliminary experiments showed that using the Fourier descriptor representation has no noticeable effect on the contour grouping performance relative to using simple up-scaling.



(a) (b) (c)

**Figure 7.1: Issues with spatial cue suggested in the Multiscale algorithm by [10]**- The paths in (a) and (b), with green representing detected line segments and red representing the links between them, have exactly the same distance to the prior (in black) if the distance of the points on the links to points on the prior are ignored. The paths in (a) and (c) have almost the same distance to the prior if the second term in Equation 7.1 is ignored.

obtained by the EG algorithm at a coarse scale are up-scaled and used as spatial priors for grouping at fine scale (here we use $K = 5$). Each spatial prior polygon is used separately in the search at fine scale. Let us define the *spatial cue*, denoted as $f_s$, as the modified mean distance, as defined in equation 5.5, between each path $c_k$ and the spatial prior polygon $S$:

$$f_s(c_k) = \epsilon(c_k, S) = \frac{1}{2}\left(D_S(c_k) + D_{c_k}(S')\right) \tag{7.1}$$

Recall that we use this distance measure, rather than the perceptually validated CM measure, for faster computation time needed to compute the thousands of paths considered during the search.[1]

Note that the above cue is different from the spatial cue suggested in [10], which ignored the distance of the points on the links between the line segments to the prior (compare Figure 7.1(a) and 7.1(b)) as well as the distance of the points on the prior to the path (the second term in equation 7.1; compare Figure 7.1(a) and 7.1(c)). Also, in

[1]The distance values between the spatial prior and all lines and links can be pre-computed and used for calculation of the distance between the spatial prior and sequences constructed from these lines and links. This greatly speeds the computation.

their method, a probability value was assigned to each line segment given its distance to the spatial prior and used in a Markov chain model. In our method, distance to the prior is used as a global cue, resulting in a probability value being assigned to the path without the Markov assumption.

Although using more priors increases the probability of having better spatial priors, as suggested by the ranking curves in the previous chapter, often a limited number of spatial priors are used in practice due to time complexity. In addition, to lower the number of closed contours produced in each of the $K$ multiscale runs, we use a memory budget equal to $1/K$ times the size of memory budget used for the single scale runs (i.e. the runs without any spatial prior) to lower the number of closed contours produced, allowing the final ranking stage to be feasible. Despite this limitation, we will show that the multiscale runs can perform better than the single scale run.

### 7.1.1 Prediction of path errors given spatial prior

To learn error predictors given multiscale priors, only the best of top 5 contours (i.e. the one with the lowest normalized CM error) obtained at the coarse scale for each of the 30 training images is used. Only the ground truth contour represented by this prior (i.e. the ground truth contour with the smallest normalized CM distance to the contour) is used in evaluating the error of the open paths to provide training samples for learning nonparametric regression models.

We train models for a combination of three cues (local, global color, and spatial priors) in two iterations. In the first iteration, we run the contour extraction algorithm, setting the cost of paths by equation 5.6 based on local and global color cues, given the models learned in 5.2.3. We use the paths obtained in this iteration to learn an initial regression model for combining the three cues. This initial model is used in a second run of the contour extraction algorithm, using all three cues in determining the cost of paths. We use the paths obtained in this second run to learn the final regression

model (see Figure 7.2). These models suggest lower average path errors when using the spatial prior as a cue, and also a roughly linear relationship between the error of the path and its distance to the spatial prior, confirming the benefit of using the multiscale priors.

As in section 5.2.3, we take a nonparametric approach to the regression problem, binning the spatial cue $f_s(c_k)$ and then estimating the mean $\hat{\epsilon}_s$ and variance $\sigma_s^2$ of predicted error in each bin. The same procedure is followed for local and global color cues. Assuming independence between the three types of cues and normal distribution of error values at each cue bin, the least-squares error prediction $\epsilon_{ML}$ is then given by

$$\hat{\epsilon}_{ML} = \frac{\hat{\epsilon}_l/\sigma_l^2 + \hat{\epsilon}_g/\sigma_g^2 + \hat{\epsilon}_s/\sigma_s^2}{\sigma_l^{-2} + \sigma_g^{-2} + \sigma_s^{-2}} \tag{7.2}$$

This predicted error is used as the multiscale version of the path cost in step (d) of the closed contour extraction algorithm (Section 5.1).

### 7.1.2 Effect of the multiscale prior

Figure 7.3 shows the minimum achievable CMnorm error among the 3 sets of closed contours mentioned above and their unions, averaged over the validation set.[1] Although *SS-Fine* has a higher average error than *SS-Coarse*, the multiscale prior improves performance at fine scale, as can be seen by comparing the average error of *MS-Fine* versus the average error of *SS-Coarse* and *SS-Fine*. Also, the effect of the multiscale method on lowering the combined error in the fine scale, i.e. *SS-Fine* $\cup$ *MS-Fine*, denoted as *2+3* is shown. This combination results in a statistically significant drop in the average error compared with any of the 3 sets as shown in Table 7.1.

By also combining the results of the coarse scale, an even lower minimum error can be achieved. Combining the results obtained from running the single scale algorithms

[1] The closed contours are all up-scaled to the original image resolution in SOD when measuring the CMnorm error.

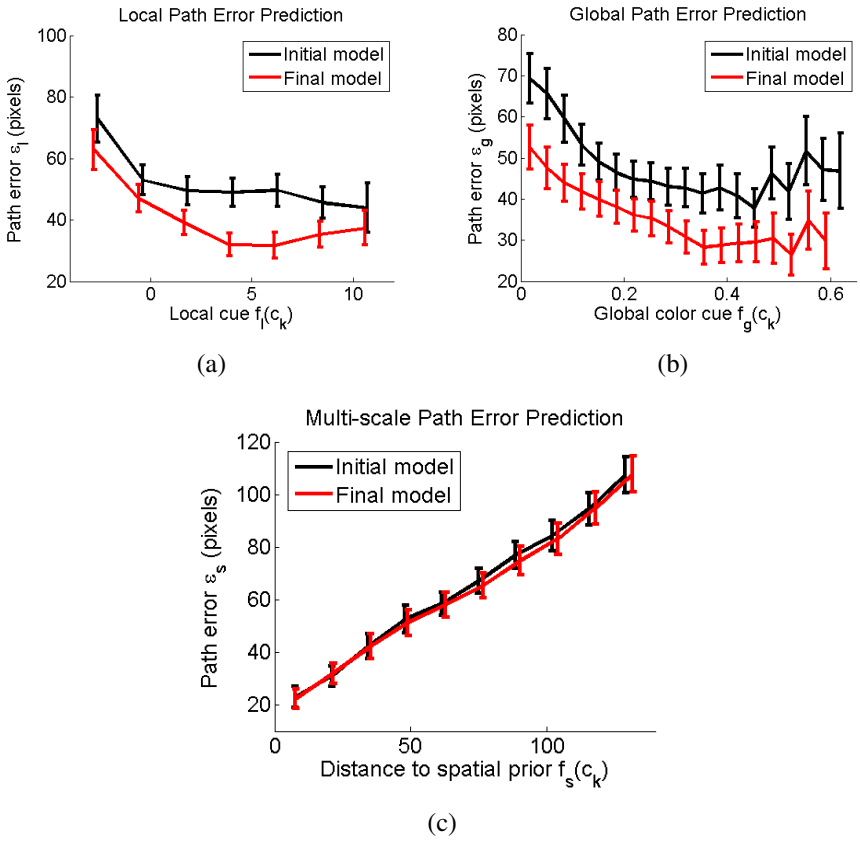



**Figure 7.2: Learned nonparametric predictors for multiscale path error** based upon (a) local, (b) global color, and (3) spatial prior cues. Error bars indicate standard error of the mean among models learned for distinct objects.

in the coarse and fine scale, i.e. *SS-Coarse* ∪ *SS-Fine* denoted as *1+2* in the figure, yields improvement over either scale alone. Combining all three sets, as shown in the figure by *1+2+3* referring to the union of the 3 sets of hypotheses, i.e. *SS-coarse* ∪ *SS-Fine* ∪ *MS-Fine* is significantly better than *1+2*, and emphasizes the effect of the multiscale prior.

A fourth alternative is to only consider the results of the fine scale with multiscale prior together with the coarse scale, i.e. *SS-coarse* ∪ *MS-Fine*, denoted in Figure 7.3 as *1+3*. This set has almost the same quality as using all 3 sets of hypotheses, but requires less computation. There is no statistically significant difference between this set and

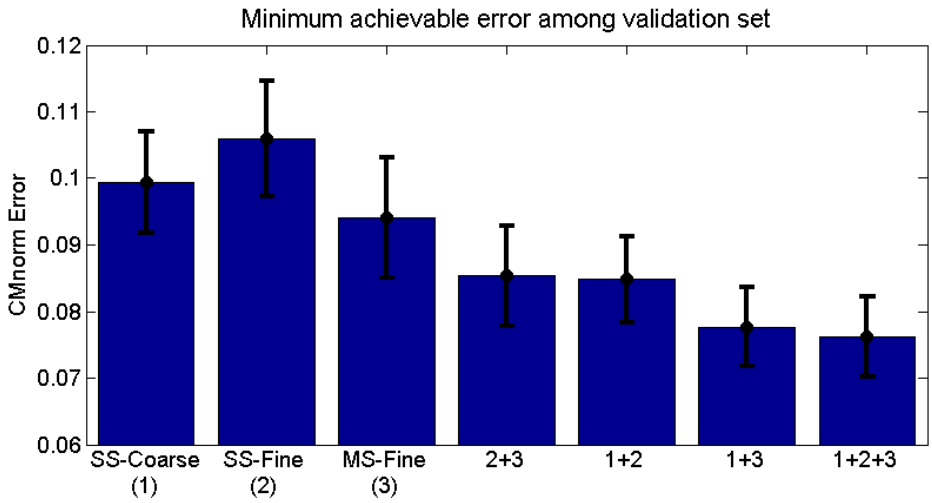



**Figure 7.3: Effect of the multiscale prior on the validation set**. The plot shows the minimum CMnorm error among all contours in the sets SS-Coarse, SS-Fine, MS-Fine, as well as their combinations. Error bars indicate standard error of the mean.

the union of all 3 sets, as shown in Table 7.1.

Figure 7.4 shows the number of closed contours among the 3 sets of closed contours mentioned above and their combinations. While the memory budget for each search using a spatial prior was only 20% of the budget for search without a prior, the total output of closed contours over all 5 priors was on average 15% higher than without a prior.

The effect of the multiscale prior in the constructive phase is obviously dependent

| p-value | SS-Coarse (1) | SS-Fine (2) | MS-Fine (3) | 2 + 3 | 1 + 2 | 1 + 3 |
|---|---|---|---|---|---|---|
| SS-Fine (2) | 0.5465 | | | | | |
| MS-Fine (3) | 0.3677 | 0.0765 | | | | |
| 2 + 3 | **0.0417** | **1.13e-5** | **0.0427** | | | |
| 1 + 2 | **4.04e-4** | **4.75e-5** | 0.2013 | 0.8988 | | |
| 1 + 3 | **3.98e-5** | **1.45e-6** | **0.0115** | **0.0377** | **0.0059** | |
| 1 + 2 + 3 | **1.44e-5** | **1.63e-7** | **0.0061** | **0.0100** | **0.0003** | 0.0589 |

**Table 7.1: p-values for pairwise repeated measures t-tests across combinations of closed contour sets as shown in Figure 7.3**- The drop in minimum achievable error using multiscale prior is statistically significant at $\alpha < 0.05$ level. The drop between using all 3 sets versus using 1+3 is not statistically significant at $\alpha < 0.05$ level.

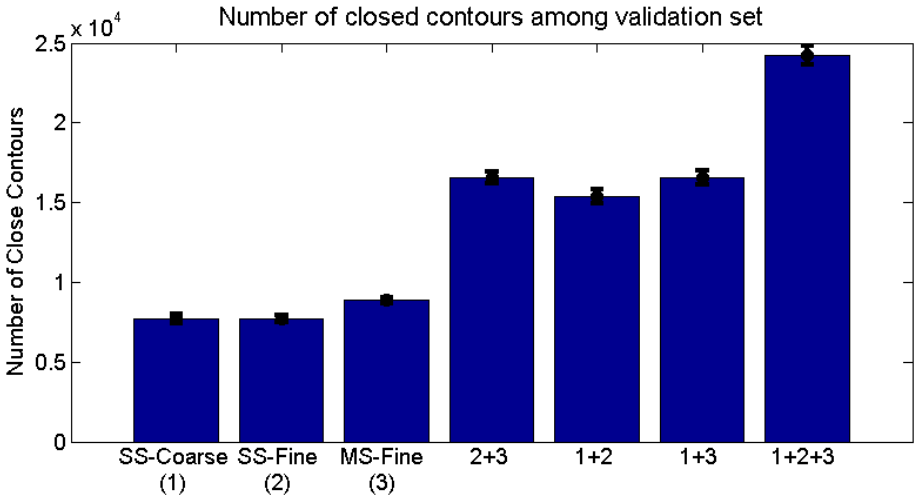



**Figure 7.4: Number of contours among sets of closed contours**- Error bars indicate standard error of the mean.

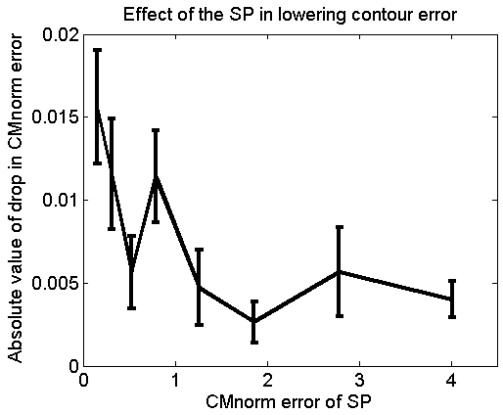



**Figure 7.5: Effect of the multiscale prior given its quality**. The plot shows the absolute value of the drop in CMnorm error obtained by using multiscale prior versus the CMnorm error of the multiscale prior, averaged over 5 spatial priors and all objects in validation set. Error bars indicate standard error of the mean.

on the quality of the spatial prior. If the spatial prior has a high error relative to the ground truth, it will not result in significant improvement relative to the coarse scale. Figure 7.5 show the relationship between the error of the multiscale prior versus its effect on lowering the minimum error among all contour hypotheses, averaged over 5 multiscale priors and all objects available in 40 validation images. This figure shows that any improvement in the quality of grouping at the coarse scale would lead to better performance in a multiscale framework.

Figure 7.6 shows examples of closed contours produced using the multiscale prior. The first row shows ground truth contours from SOD. The best contours found in the coarse and fine scale without using the multiscale prior are shown in the second and third rows respectively. Using the spatial prior contours selected from the top 5 ranked contours in the coarse scale, shown in the fourth row, a new set of contours in the fine scale is obtained, the best of which are shown in the fifth row. As can be seen, the multiscale prior helps in finding details without getting distracted in clutter. I believe that the spatial prior helps by i) focusing the greedy search on the region containing the salient object, and ii) by encouraging contours that are compact and smooth.

In some cases the contours found by the single-scale algorithm at fine scale (*SS-Fine*) are better than or the same as those found by the multiscale algorithm (*MS-Fine*): the multiscale prior does not improve the results. In Figure 7.7 for example, the multiscale prior seems to be distracting the search, instead of guiding it.

## 7.2 Spatial prior in the ranking phase

A different approach to using the spatial prior in a grouping algorithm is to use it in the ranking phase to choose the closed contours that best resemble the prior. Depending on the quality of the prior, information about location, smoothness, color distribution, etc. can be used to evaluate the set of closed contour hypotheses. We explored using the distance to the spatial prior (Equation 7.1) as a ranking cue in addition to the 4 cues we

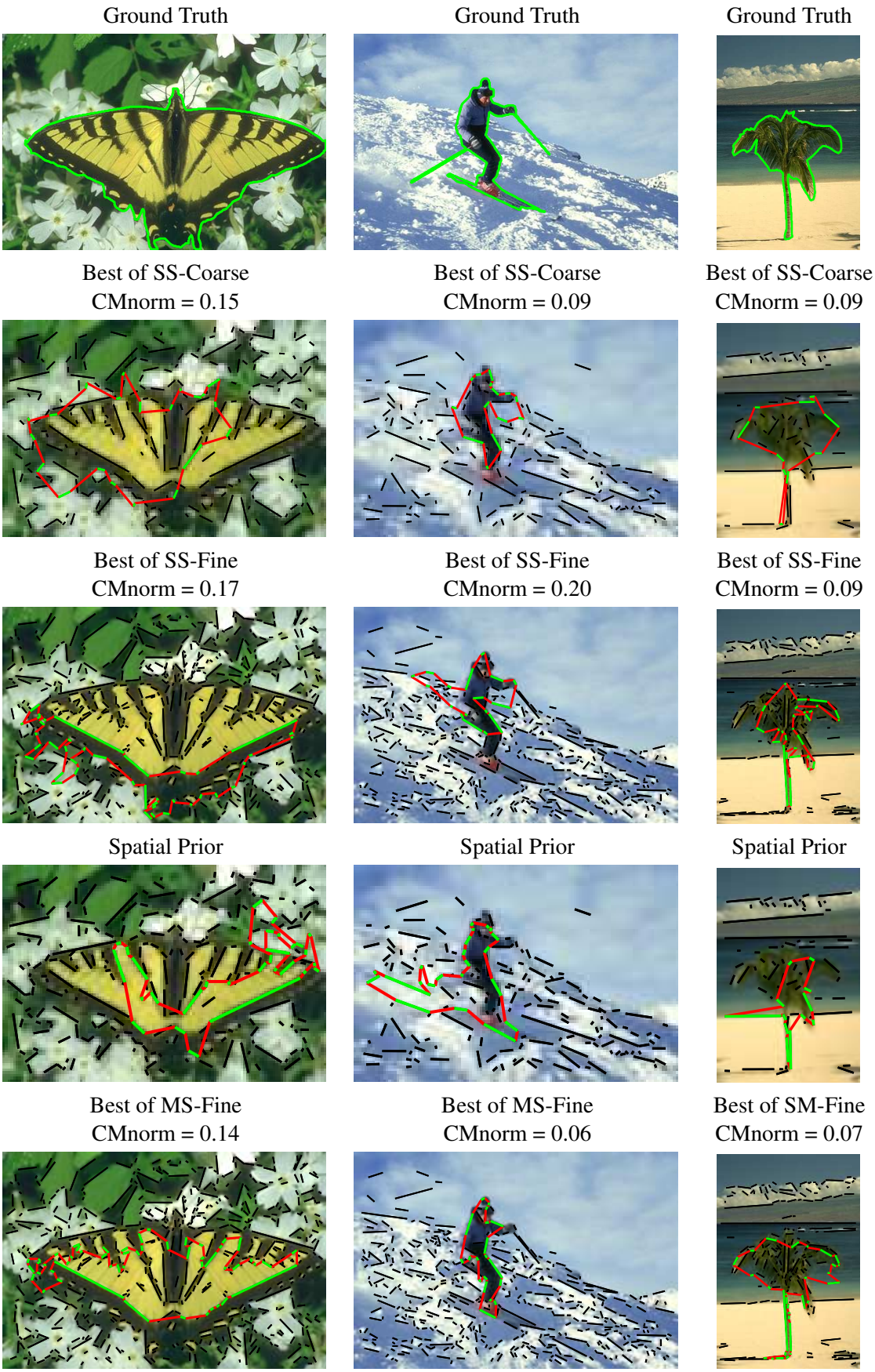



**Figure 7.6: Example output of the multiscale algorithm**. First row, ground truth contour from SOD; second row, best contour produced by the single-scale algorithm at coarse scale; third row, best contour produced by the single-scale algorithm at fine scale; fourth row, a spatial prior contour from coarse scale; and fifth row, best contour produced by the multi-scale algorithm at fine scale.

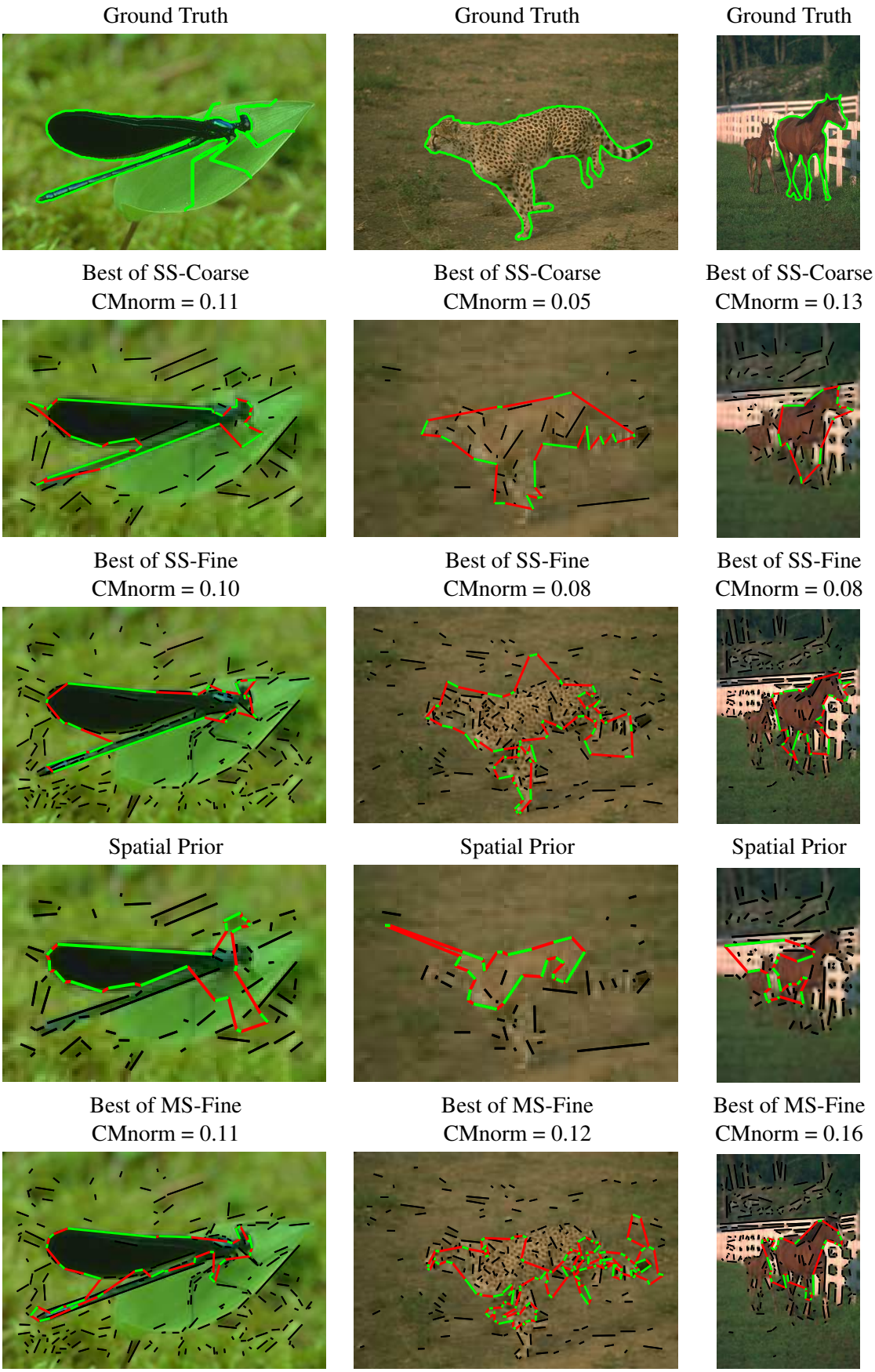



**Figure 7.7: Example output of the multiscale algorithm- cont.**. First row, ground truth contour from SOD; second row, best contour produced by the single-scale algorithm at coarse scale; third row, best contour produced by the single-scale algorithm at fine scale; fourth row, a spatial prior contour from coarse scale; and fifth row, best contour produced by the multi-scale algorithm at fine scale.

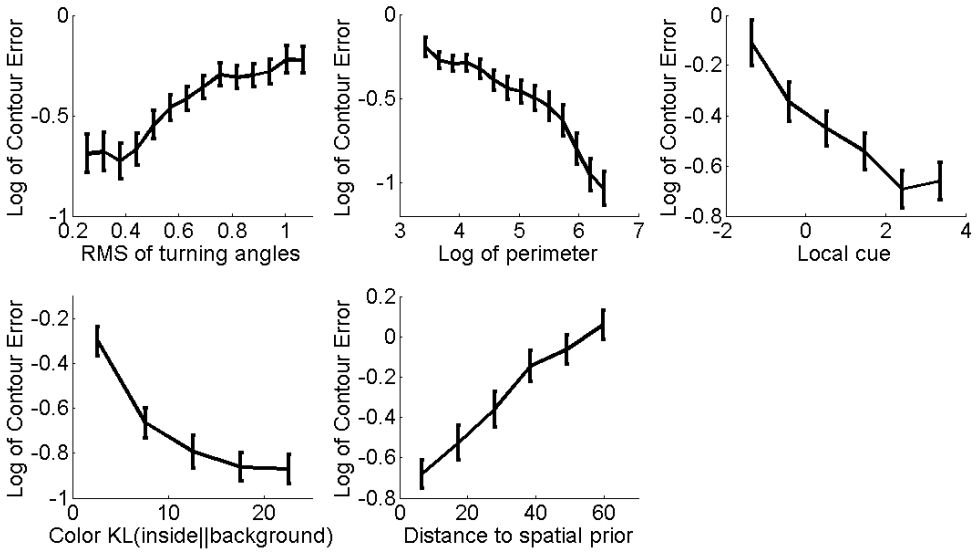



**Figure 7.8: Learned nonparametric error predictors of closed contours at fine scale**- Error bars indicate standard error of the mean among models learned for distinct objects.

used for ranking in Section 6.1. The minimum distance with respect to the 5 multiscale priors is used. The regression models learned for ranking closed contours produced at the fine scale using this cue together with the 4 cues introduced in the previous chapter are shown in Figure 7.8.

The performance of this ranking algorithm is shown in Figure 7.9 for the validation set. In this figure, ranking of 3 sets of contours with and without the spatial prior cue are compared. The three sets are *SS-Fine*, *MS-Fine*, and their union, *Fine*. The results without using the spatial prior cue in ranking are shown as dashed curves, while results with the spatial prior cue are shown as solid curves. It is clear that using distance to the multiscale priors can help in improving the performance of ranking and hence the performance of the grouping algorithm. In addition, this figure shows that we get better performance by just considering the contours in *MS-Fine* and ignoring contours generated in *SS-Fine*, without the prior.

We employ the diversity method described in Section 6.2, but with a stricter criterion: contours with more than 20% overlap are considered redundant; i.e. $\alpha = 0.2$.

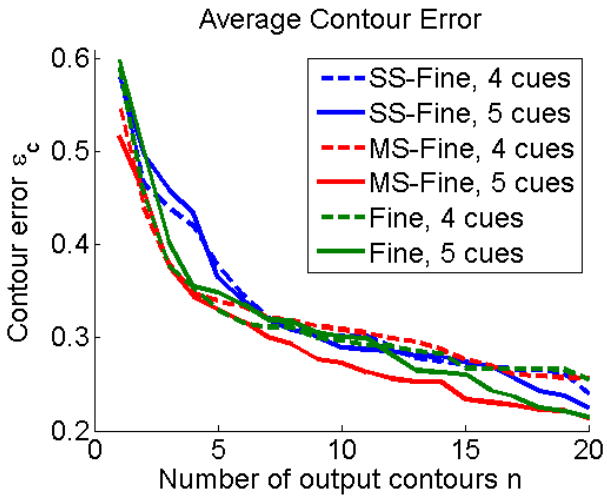



**Figure 7.9: Ranking performance at fine scale**- Average error on the validation set.

The stricter criterion is necessary, as the multiscale contours have a higher similarity to each other due to the spatial prior. This value for $\alpha$ is selected by optimizing performance on the validation set. In rare cases where applying such a strict value on dissimilarity results in too few contours being selected, $\alpha$ was increased in increments of 0.05 until at least $n = 20$ output contours were left for selection.

## 7.3 Combining Coarse and Fine Results

So far in this chapter, I have discussed the use of the multiscale prior in improving the results at the fine scale, both in the constructive phase and the ranking phase. The set of contours obtained using the multiscale prior, *MS-Fine*, was shown to yield lower error than *SS-Fine*, and combining the two was found to yield no improvement in performance. Here we ask whether combining *MS-Fine* contours with *SS-Coarse* contours detected using our single scale algorithm at coarse scale may improve performance even further.

Note that both these sets are potentially valuable. Depending on the quality of the multiscale priors and the level of detail, clutter and noise in the image, either the

set of contours produced at coarse scale or those at fine scale may contain the best approximation to the salient object in the image. In addition, the size of the salient object in the image can affect the scale at which the object can best be described. It is therefore better to choose the top contours from both of these sets.

Given two sets of $n$ selected contours from each of the two scales, the errors predicted by the ranking models can be used to sort them and choose only $n$ contours from the two sets. The performance of this system fusing coarse and fine contours is shown for the training and the validation sets in Figures 7.10(a) and 7.10(b) respectively. We made the following observations:

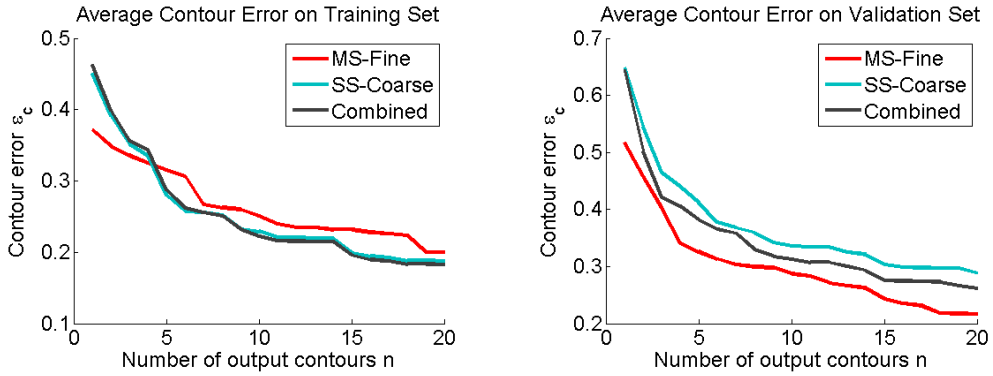
1. The inconsistency in the results for training and validation sets suggests that we have insufficient data.

2. For both coarse and fine scales, the predicted errors are systematically different from the actual errors. Figure 7.11(a) shows the actual versus predicted errors and their deviation from the ideal prediction in the training set. Although the mean of predicted errors are very close to the mean of actual errors, the variances are very different. Our hypothesis is that this may be due to inappropriate application of the cue combination rule. In particular, while this model should apply when the predicted errors are unbiased estimators of the true error, they will in fact often be biased.

As future work, we suggest using (i) a much larger training dataset, and (ii) a cue combination method that does not suffer from the above issue, for example, the following simple linear regression model for combining the error predictions by each cue:

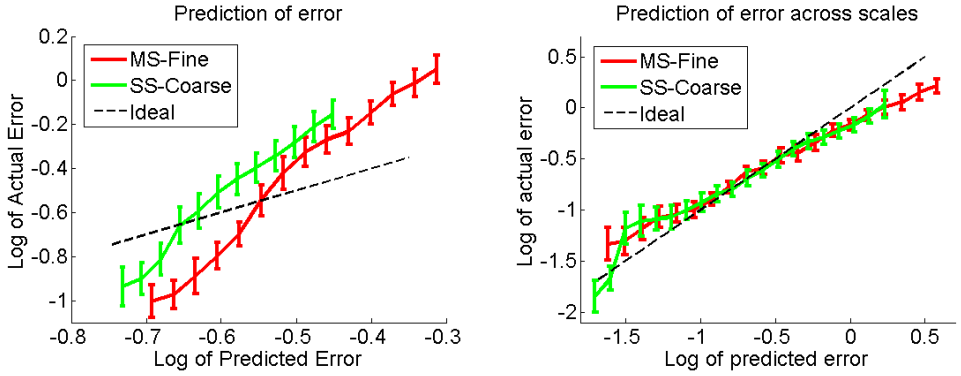
$$\hat{\epsilon} = w_0 + \sum_i w_i \hat{\epsilon}_i. \tag{7.3}$$

Figure 7.11(b) shows the actual error versus the predicted error in the training set using the above equation, with weights learned from the training data. Although this simple




(a) (b)

**Figure 7.10: Ranking performance of combined coarse and fine contours**- Average error on (a) the training set and (b) the validation set.




(a) (b)

**Figure 7.11: Actual versus predicted errors in coarse and fine scales**. (a) Due to the failure of our assumptions, our predicted error (Equation 6.4 ) deviates from the actual error for those that are not close to the average error; (b) Using a simple linear regression method (Equation 7.3) can improve combination of contours across scales. Error bars indicate standard error of the mean among distinct objects.

method does not improve our ranking results for each scale or their combination, it greatly improves the accuracy of the error prediction.

## 7.4 Conclusion

In this chapter, methods for using spatial priors in the grouping algorithm were discussed. Multiscale priors from the coarse scale were used in i) a guided search for closed contours at the fine scale, and ii) for ranking closed contours found at the fine scale, showing improvements in performance at fine scale. However, when combining coarse and fine contours, our error prediction often favours coarse contours over fine contours, even when they have higher actual errors. We suggest using a much larger training dataset and improving the error prediction method as future work for improved results.

In the next chapter, I will compare the performance of our grouping algorithm with other grouping methods in the literature.

# 8

# Experiments

In this dissertation, we have shown a contour grouping method that benefits from combining local and global cues, as well as multiscale priors. We also introduced a salient object dataset (SOD) and a suitable evaluation measure (CM) for evaluating contour grouping methods. In this chapter, I will compare the results of our proposed method against some existing grouping methods, using the evaluation tools proposed. I will first introduce the test set and review the methods of evaluation. Finally, I will present and discuss results.

## 8.1 Test set

The test set consists of 61 images from the SOD dataset. These images are a subset of 100 images reserved as the test set in the Berkeley Segmentation Dataset (BSD) [3]. However, not all these 100 images are suitable for testing grouping methods. Therefore, a subset of 61 images including at least one clear object entirely contained within the image boundaries was selected. This set is a superset of the test set used in [10] which contained only 20 test images. A few samples from the test set are shown in Figure 8.4.

## 8.2 Experiment settings

We evaluate two versions of our method: i) Enhanced Grouping (EG): Our grouping method at a single (coarse) scale, and ii) Multiscale Enhanced Grouping (MEG): Our grouping method applied at two scales, using coarse priors at the fine scale.

To match training, test images were down-sampled by a factor of 4 prior to running our EG method, and also by a factor of 2 for running the fine scale in MEG. Resulting closed polygons were then up-sampled to match the original resolution for evaluation.

Two error measures were used for evaluation:

i) The region-based error measure based on the popular intersection to union measure, defined previously in Equation 3.2 as

$$\text{RI}(A, B) = 1 - \frac{|R_A \cap R_B|}{|R_A \cup R_B|}$$

where $A$ denotes the algorithm contour, $B$ the ground truth contour, $R_A$ and $R_B$ the the set of pixels interior to the algorithm and ground truth boundaries respectively. Also $|X|$ measures the number of pixels in the set X.

ii) A contour based error *CMnorm* based on a normalized version of the contour mapping (CM) measure [50] defined as the average distance between corresponding pixels on contours $A$ and $B$ (Equation 3.11), normalized by the square root of the area of ground truth boundary $B$, i.e. $\sqrt{|B|}$. The CM measure determines the monotonic mapping that minimizes this distance, and as shown in Chapter 3, it accurately captures human judgements of shape segmentation error [50].

Recall that for each image in the SOD dataset, there are a number of salient ground truth object boundaries generated by multiple human subjects. The error of an algorithm-generated contour is defined as the *minimum* error over all ground truth contours for

that image. In addition, when algorithms are allowed to report multiple contours per image, we report the error as the *minimum* error over all algorithm contours for the image, as is the norm (e.g., [20]).

## 8.3 Competition

We compare our grouping algorithms with four previous approaches:

1. The Regional Ratio Contour (RC) algorithm of Stahl & Wang [19],
2. The Adaptive Grouping (AG) method of Estrada & Jepson [27],
3. The Multiscale (MS) method of Estrada & Elder [10], and
4. The Superpixel Closure (SC) method of Levinshtein & Dickinson [20].

Over the vast literature on contour grouping, these are the most prominent grouping methods that compute *closed* object boundaries. Note that nothing is declared about the rest of the image pixels, as they may belong to other salient objects or background, therefore these methods do not perform complete segmentation of images, but are *salient object segmentation* methods.

All of the methods except SC start with edge detection and line approximation, and the goal is to group the line segments into a sequence representing a simple and closed boundary. SC groups superpixels and not edge fragments. However, this method uses contour-based cues and is in many ways similar to other contour grouping methods. The other 3 methods all use local grouping cues such as proximity and good continuation. All methods except for the RC method use colour cues in some way. MS is unique in its use of a coarse-to-fine framework. Figure 8.1 summarizes the main differences among these grouping methods.

All of the above four grouping methods were evaluated at full resolution, as in the original papers, using the authors' codes and recommended parameter settings. For the

| | | RRC<br>Stahl & Wang | AG<br>Estrada & Jepson | MS<br>Estrada & Elder | SC<br>Levinshtein & Dickinson | MEG<br>Movahedi & Elder |
|---|---|---|---|---|---|---|
| Prep | Edge detection | Canny | Canny | Elder's | Global Pb | Elder's |
| | Line detection | Kovesi | Jepson's | Elder's | N/A- Instead applies superpixels Segmentation | Elder's |
| **Unary** | Color contrast | Not used | Not used | At fine scale- RGB | Indirectly, by GlobalPb and superpixels | Not as a unary cue, but as a *global* cue |
| | Texture contrast | Not used | Not used | At fine scale | Indirectly, by GlobalPb and superpixels | Not used |
| | Brightness contrast | Not used | Not used | At coarse scales | Indirectly, by GlobalPb and superpixels | Not used |
| | Pb (Martin's) | Not used | Not used | Yes | GlobalPb used instead | Not used |
| **Binary** | Proximity | Yes, as gap in cost function | Yes, an exponential function of gap | Yes, as likelihood of gap | Yes, as gap in cost function | Yes, as one of the cues in the mixture model |
| | Smooth Continuation | Not used, argues does not help | Yes, as a heuristic function of the angles | Yes, as likelihoods of parallelism and collinearity | Indirectly, by superpixels | Yes, as parallelism and collinearity cues in the mixture model |
| | Normalization | Not used | Normalized by total affinities of possible extensions | Not used | Not used | Not used |
| | Color similarity/dissimilarity | Not used | Yes, histogram similarity | Not used | Indirectly, by GlobalPb and superpixels | Not as a binary cue, but as a *global* cue |
| | Brightness & contrast similarity | Not used | Not used | Yes | Indirectly, by GlobalPb and superpixels | Yes, as a local cue in the mixture model |
| **Global** | Area enclosed | Yes | Not used | Not used | Yes | Yes, as a ranking cue |
| | Compactness | Indirectly, by total gap/area cost | Discard any below a threshold | Indirectly, by multiscale prior | Indirectly, by superpixels | Yes, as RMS of turning angles |
| | Multiscale prior | Not used | Not used | Yes, as local distance/ angle to spatial prior | Not used | Yes, as a global cue both in greedy search and ranking |
| | Closure | Yes | Yes | Yes | Yes | Yes |
| | Self intersection (Simplicity) | Detected & removed | Detected & removed | Detected & removed | N/A to superpixels | Detected & removed |
| Search & Saliency | Search Algorithm | Ratio Contour Algorithm | Greedy search | Greedy search | Parametric maxflow | Greedy search |
| | Saliency / cost | Total gap divided by area of region | Color homogeneity of inside and outside of contour | Posterior probability OR geometric mean | Total gap divided by area of region | Error prediction |

**Figure 8.1: Comparison of contour grouping methods**

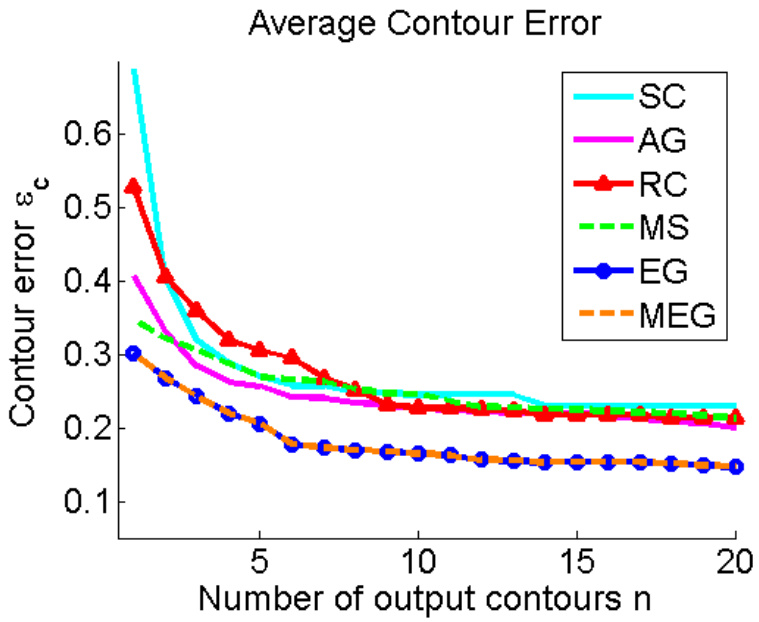
preprocessing stage of the Regional Ratio Contour algorithm, Canny [71] edge detection and Kovesi [84] line detection methods were applied, as suggested by the authors. For the Adaptive Grouping method, Canny [71] edge detection and Jepson's robust line detection [85] methods were used, as suggested by the authors. For MS, Elder and Zucker's edge detection [73] and line approximation [74] methods were used, as in our method, but with parameters suggested by the authors. The SC method was run using 200 superpixels obtained using global Pb edge maps [86] with a threshold value of $T_e = 0.05$, as suggested by the authors.

## 8.4 Comparative Results

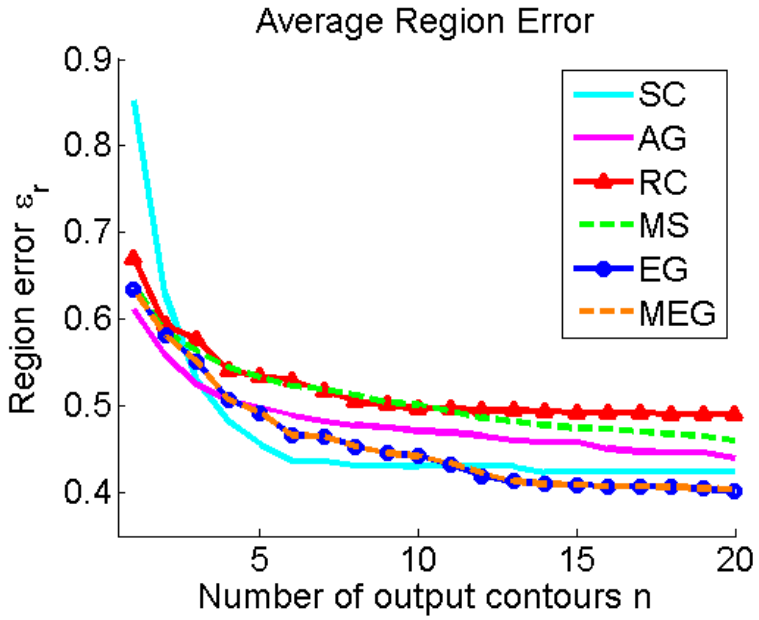
### 8.4.1 Quantitative Results

Figure 8.2 shows quantitative results. In panels (a) and (b) we report the average contour error $\epsilon_c$ and region error $\epsilon_r$, respectively, as a function of the number of contours $n$ the algorithm is allowed to report. We see that by the contour measure, our EG algorithm outperforms all methods. Also MEG provides almost exactly the same results as EG. By the region measure, our EG method outperforms all other methods for large $n$ and is better than all except the AG method for small $n$, and the SC method for $4 \leq n \leq 10$.

Figure 8.3(a) shows how the best contours computed by each method compared, using the contour error measure $\epsilon_c$. This result suggests that refinements to the method for ranking computed closed contours could lead to further gains for small-$n$ evaluation. Here the superiority of the EG and MEG method derives in part from the larger number of contours computed (Figure 8.3(b)).

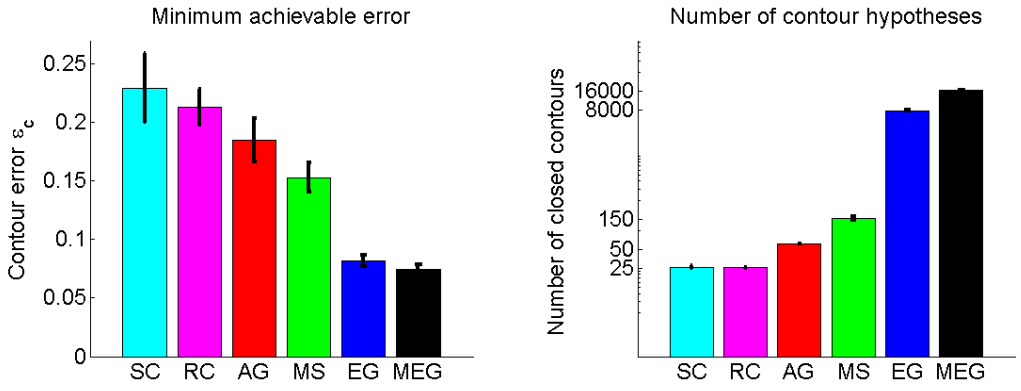



(a)




(b)

**Figure 8.2: Quantitative evaluation on SOD test dataset- Top 20 contours**. (a) Minimum contour error using contour mapping measure as a function of the number of output contours allowed; and (b) Minimum region error using intersection over union measure as a function of the number of output contours allowed.




(a) (b)

**Figure 8.3: Quantitative evaluation on SOD test dataset- Minimum contour error over all output contours**

### 8.4.2 Qualitative Results

Figures 8.4 and 8.5 show qualitative results for each method. Here we show the best of $n = 20$ contours for each method on a selection of test images from the SOD dataset. More samples are shown in Appendix A.

As can be seen in Figure 8.4 our algorithm does a good job of selecting the main salient object, on average more often than the other methods. However there are still failures (Figure 8.5).

These figures also show the best contour over all contour hypotheses produced by the MEG algorithm. In some cases, there is clearly potential for improving the performance by improving the ranking method, possibly by using a better model or additional ranking cues. To learn better ranking models, more 'good' samples (with low errors) are needed. The current set of training contours contains many contours with high error values and less contours with low error values.

| Component | Image 1 $N_c = 125, N_f = 346$ run time (seconds) | Percentage | Image 2 $N_c = 411, N_f = 1016$ run time (seconds) | Percentage |
|---|---|---|---|---|
| Edge and Line Detection | 2.47 | 0.4% | 4.02 | 0.5% |
| Graph Construction | 3.00 | 0.5% | 13.43 | 1.8% |
| Constructive Search- Coarse | 89.29 | 13.7% | 80.87 | 10.8% |
| Constructive Search- Fine | 283.97 | 43.6% | 380.12 | 50.6% |
| Error Prediction and Ranked Selection | 269.96 | 41.4% | 270.49 | 36.0% |
| Other | 3.02 | 0.5% | 2.8 | 0.4% |
| Total | 651.72 | 100% | 751.74 | 100% |

**Table 8.1: Running time of the components of the MEG method**- shown for two sample images. Run time depends on the number of line segments detected at coarse scale (shown as $N_c$) and those detected at fine scale (denoted as $N_f$).

### 8.4.3 Run Time

The average run time of the multiscale version of our proposed method (MEG) on a 3.4 GHz Quad Core Intel CPU with 8GB of RAM with a MATLAB implementation is 10.94 minutes[1], while the coarse scale component (EG) has an average run time of 2.88 minutes (about 26% of the MEG run time). As a comparison, the MS method has an average run time of about 20 minutes on the same machine, again using a Matlab code, while the run times of the other 3 methods are all less than a minute with C code implementations.

While not yet competitive in speed with the RC, AG and SC methods, an advantage of the EG method is that it is highly parallelizeable, as at each stage of grouping, evaluation of continuation hypotheses and error prediction for closed contour hypotheses can be done independently for each contour. The running times of the different components of our method are shown in Table 8.1 for two sample images.

[1]This is the average of the run time over 61 test images.

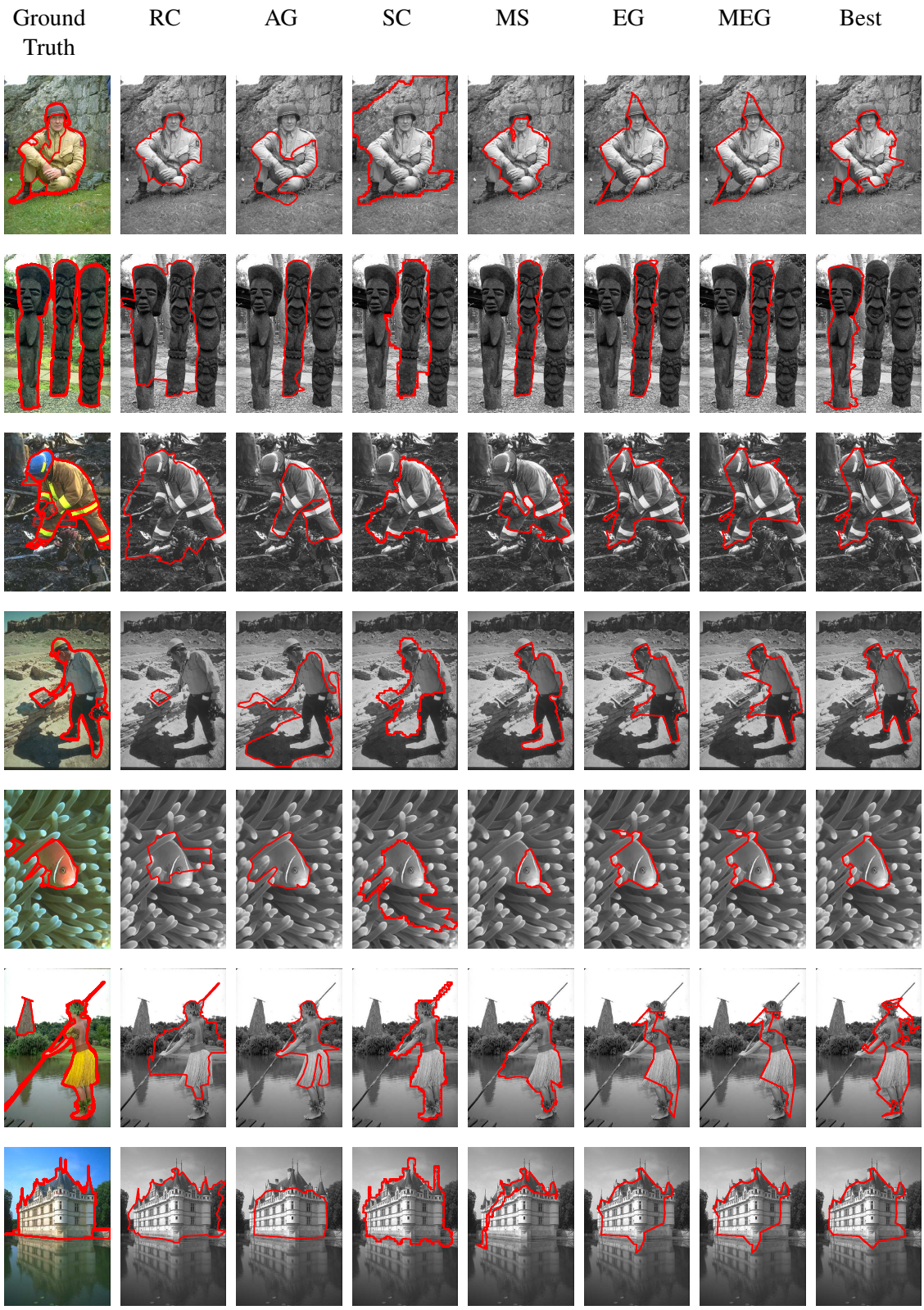



**Figure 8.4: Qualitative results (1)**- 1st column: Ground truth boundaries from **SOD**, 2nd column: best of top 20 of **RC**, 3rd column: best of top 20 of **AG**, 4th column: best of top 20 of **SC**, 5th column: best of top 20 of **MS**, 6th column: best of top 20 of **EG**, 7th column: best of top 20 of **MEG**, 8th column: best among all of MEG.

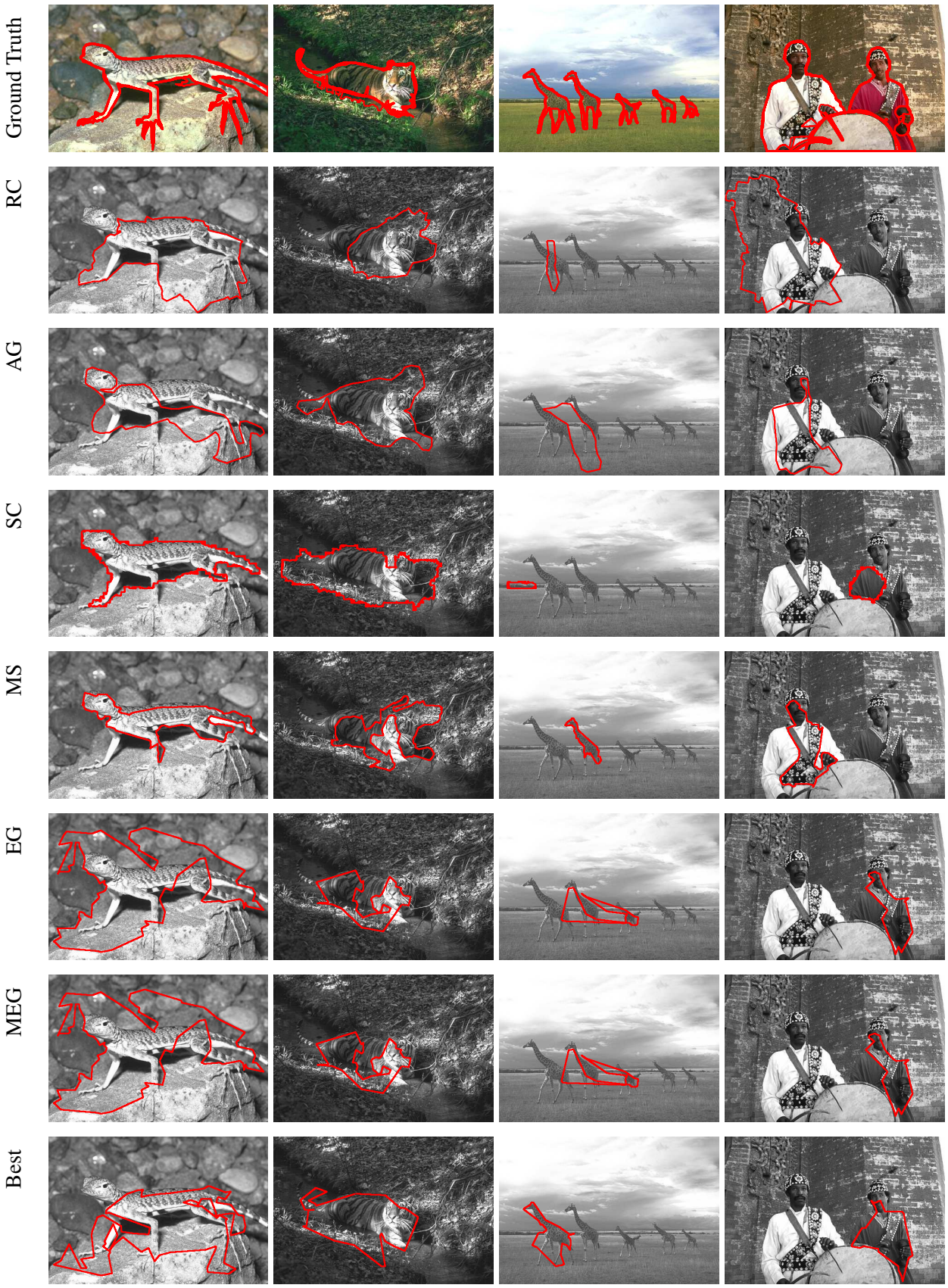



**Figure 8.5: Qualitative results (2)**- 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

# 9

# Discussion

This dissertation has focused on four main aspects of the problem of salient object segmentation:

- **Evaluation.** A dataset of ground truth object segmentations was introduced for the purpose of empirical performance evaluation of object segmentation algorithms. We considered 5 error measures and analyzed their potential weaknesses and strengths. Using psychophysical experiments, we showed that a Contour Mapping (CM) measure based upon a contour bimorphism was most consistent with human judgements. The CM measure was shown to be as predictive of human judgements as human subjects were of each other.

- **Extraction of closed contours.** We presented a novel method for combining local and global cues for improved extraction and ranking of the closed bounding contours of salient objects in natural images. Models were learned from training data for i) Construction of an association graph to model the grouping problem, ii) Constructing a framework for finding a set of closed contour hypotheses using a greedy search informed by both local and global cues within a constrained memory budget, and iii) Ranking this set of hypotheses for selection of salient contours for output.

- **Diversity.** We introduced novel algorithms for promoting the diversity of both partial and complete contour hypotheses, demonstrating substantial performance gains. When maintaining a relatively large pool of open contour fragments, diversity can be maintained by a method based on Principal Component Analysis (PCA) of data vectors that encode the line segments that each contour is comprised of, as well as its predicted error. For a smaller set of ranked contour hypotheses, however, a PCA-based method can sacrifice quality for diversity, and a method based on percentage of shared line segments results in better performance.

- **Multiscale priors.** Although Estrada & Elder [10] reported a performance improvement with multiscale priors, we were not able to get a substantial improvement using MEG versus EG. There are two possible reasons: i) The MS algorithm was tested on only 20 images, while we tested on a larger test set of 61 images; ii) We have improved the single scale algorithm. Our single scale algorithm, EG, greatly outperforms their single scale algorithm, and therefore does not equally benefit from a multiscale framework.

## 9.1 Speeding up the implementation

As mentioned before, our method is highly parallelizable, and so could be made much faster through proper implementation on a multi-core system. Currently, our method is implemented in unoptimized MATLAB code; more careful vectorization and implementation in a compiled language could result in significant speed-up. A large percentage of the running time is spent on error prediction and ranking, which requires the computation of the ranking features for each of the thousands of contour hypotheses. Parallelizing this part of the code could speed up the error prediction up to $n$ times, where $n$ is the number of processor threads. An additional improvement

could be achievable by adaptively detecting when the process of closed contour extraction can be terminated. More gain could be achieved by generalizing the contour extraction stage to allow hierarchical grouping of partial contour paths, rather than simply extending each path by a single segment at each iteration. Such a hierarchical computation could in principle lead to a logarithmic reduction in computation time, and would more closely match the hierarchical architecture of the object pathway in primate visual cortex [87, 88].

## 9.2 Future Work on Evaluation

Although the CM measure performs well, there are still some challenges that remain to be addressed in future research:

1. Mumford[68] has suggested that shape judgements are asymmetric and can depend upon context. All of the measures we consider here are symmetric, and are functions only of the segmented shapes.

2. Previous research shows the visual importance of false-negative and false-positive pixels is not necessarily the same [51, 57]. Specifically, missing object parts tend to be more important than added background. The measures we consider ignore this perceptual difference. Error values could potentially be assigned different weights depending upon whether they correspond to false negatives or false positives. These weights need to be validated by psychophysical experiments.

3. The current form of our CM measure is limited to simply-connected shapes. For example, shapes with holes cannot be compared using this measure. However the CM measure could potentially be generalized to other topologies by mapping each bounding contour separately, while enforcing topological constraints.

4. Another area for future research is the normalization of the contour mapping

measure for comparing objects of different sizes. In our psychophysical experiments, subjects were always asked to compare contours approximating the same object. However, when comparing the quality of two contours approximating two objects with different sizes, some normalization is necessary. In this dissertation, we normalized the CM measure by the square root of the area of the ground truth object, while the region-intersection measure, RI, normalizes by the area of the union of the contour region and the ground truth region. Psychophysical experiments can help us understand perceptual preferences and can therefore help in establishing normalization methods that best approximate human judgements.

## 9.3 Future Work on Contour Grouping

There is the potential for improving the performance of the proposed grouping method in the following areas:

1. The fact that the training, validation and test sets were not predictive of each other suggests that we lacked sufficient data. The performance of the grouping method can be improved by using more data.

2. The biggest potential for improving the results of contour grouping appears to be in the error prediction and ranking phase. As shown in the qualitative results of Chapter 8, the best contour available in the set of contour hypotheses is often not among the top ranked contour(s). Improving the ranking method can close the gap between the error values seen in Figure 8.2 and those in 8.3(a)[1].

   Ranking may be improved by employing more sophisticated learning methods, such as those specifically designed for ranking (e.g. [89, 90]), or by employing

[1]The best contours found by the MEG algorithm have an average error of 0.0743, while the best contours among the top 20 have an average error of 0.1400

more informative ranking features. Also as discussed in Section 7.3, the cue combination method needs to be adjusted for improved error prediction. We also believe that more data samples are needed to efficiently employ these models. Specifically, more 'good' samples with low error values are needed.

## 9.4 Other Future Work

As shown in Figure 6.3 the sum over contour hypotheses computed by our grouping method forms a saliency map. It would be interesting to compare these saliency maps with saliency maps obtained by other methods [82, 91, 92].

# Appendix A

# Qualitative Comparison

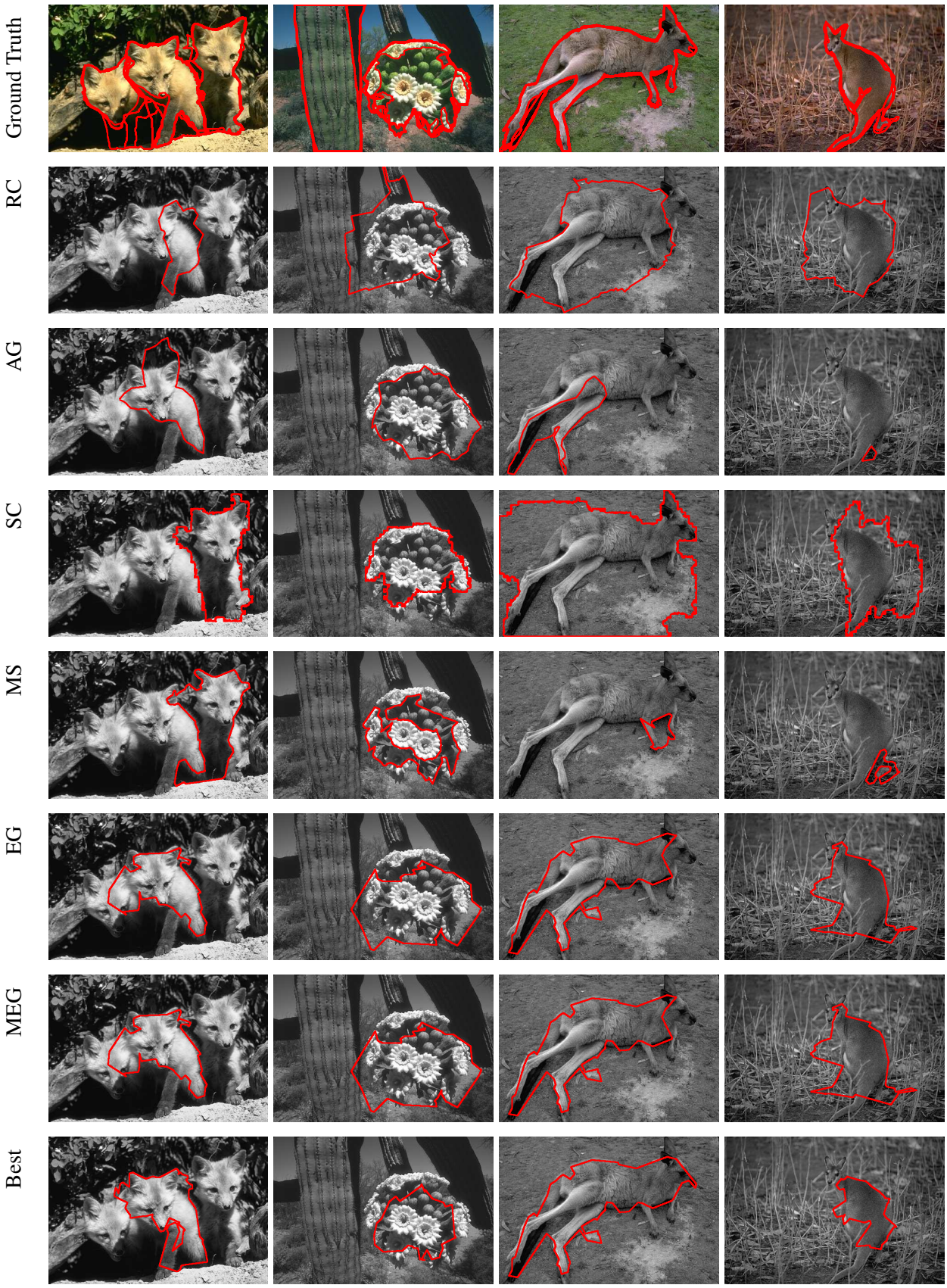



**Figure A.1: Qualitative results**- 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.


**Figure A.2: Qualitative results**- 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

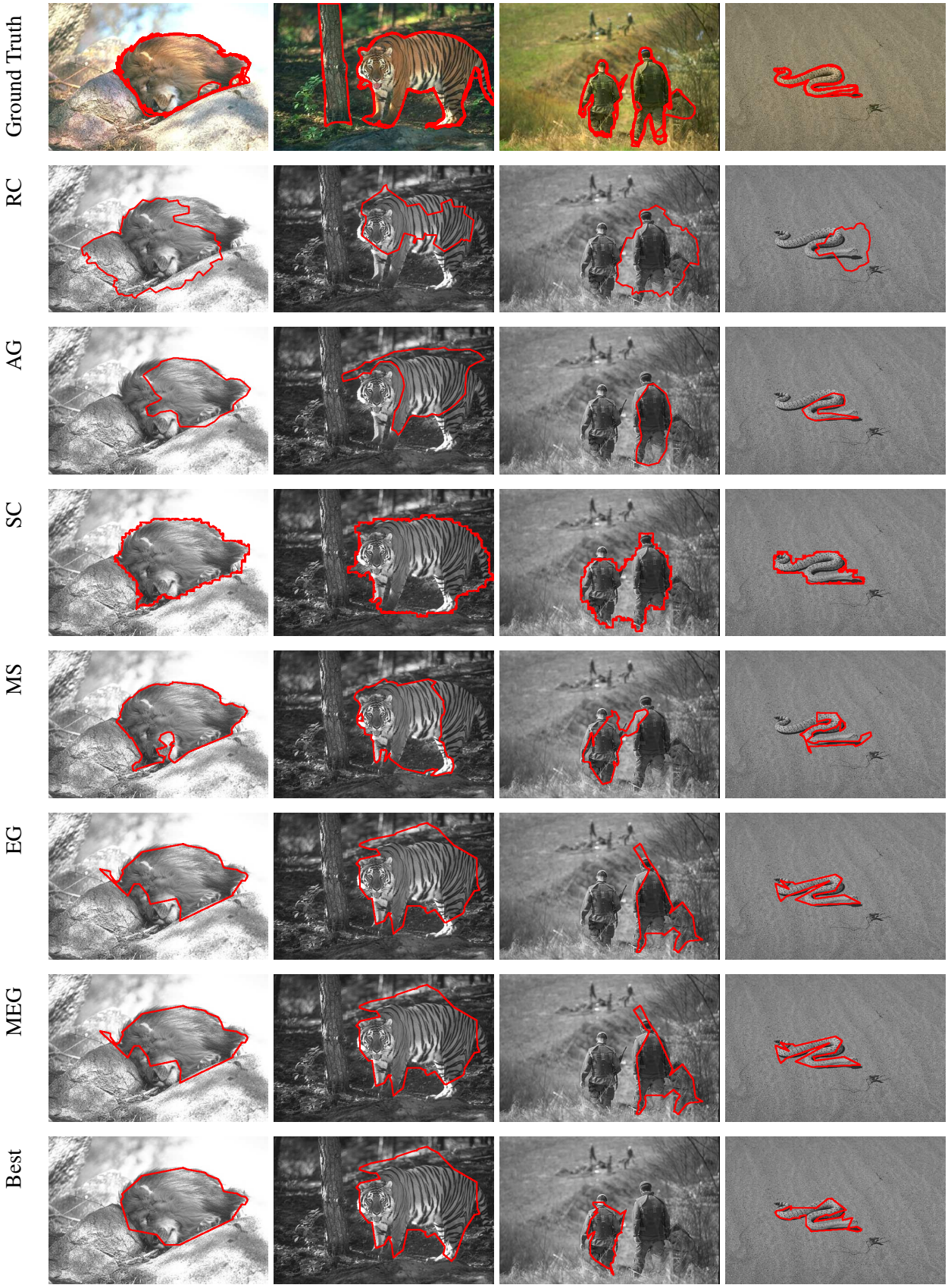



**Figure A.3: Qualitative results** - 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

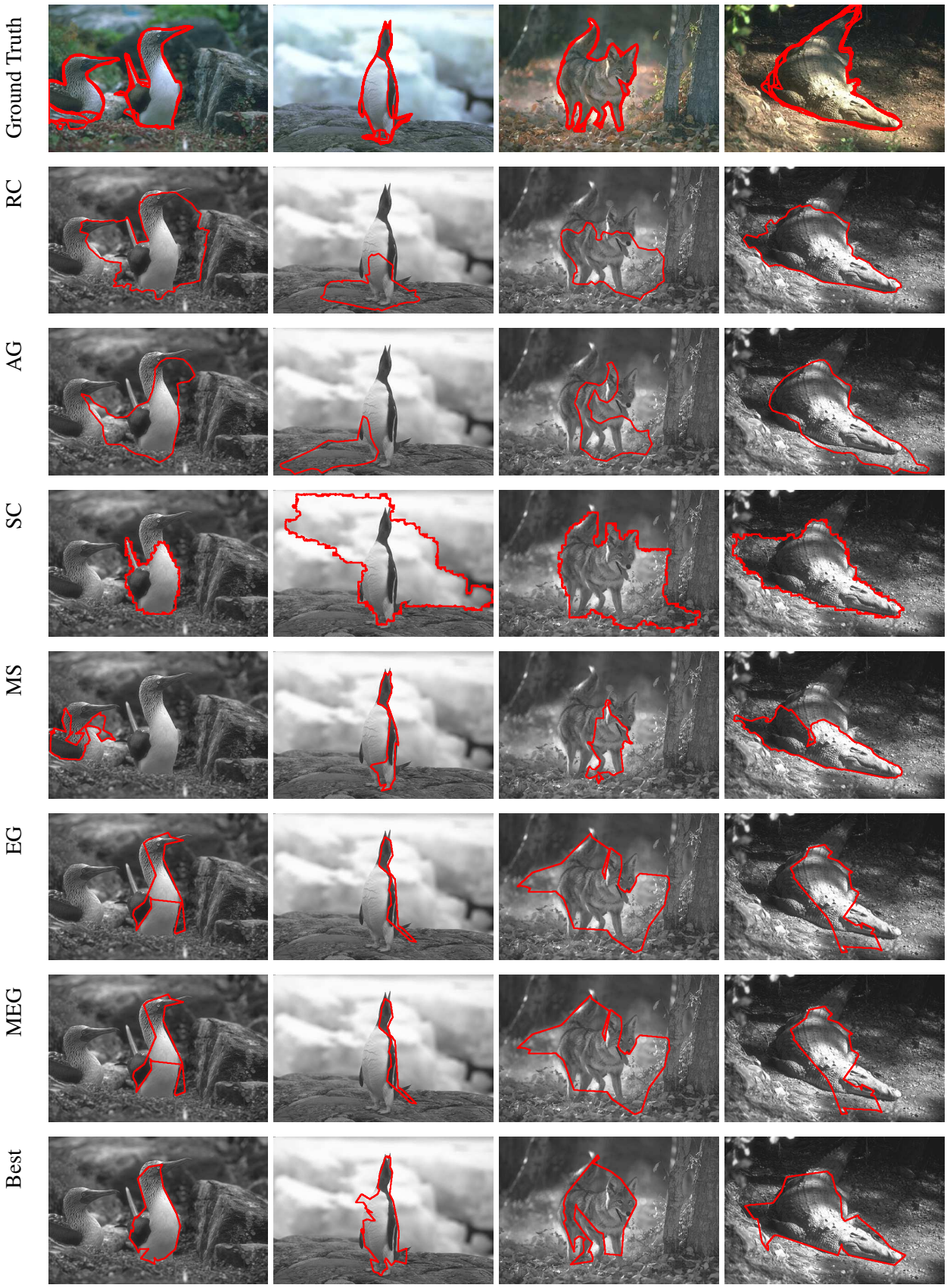



**Figure A.4: Qualitative results** - 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

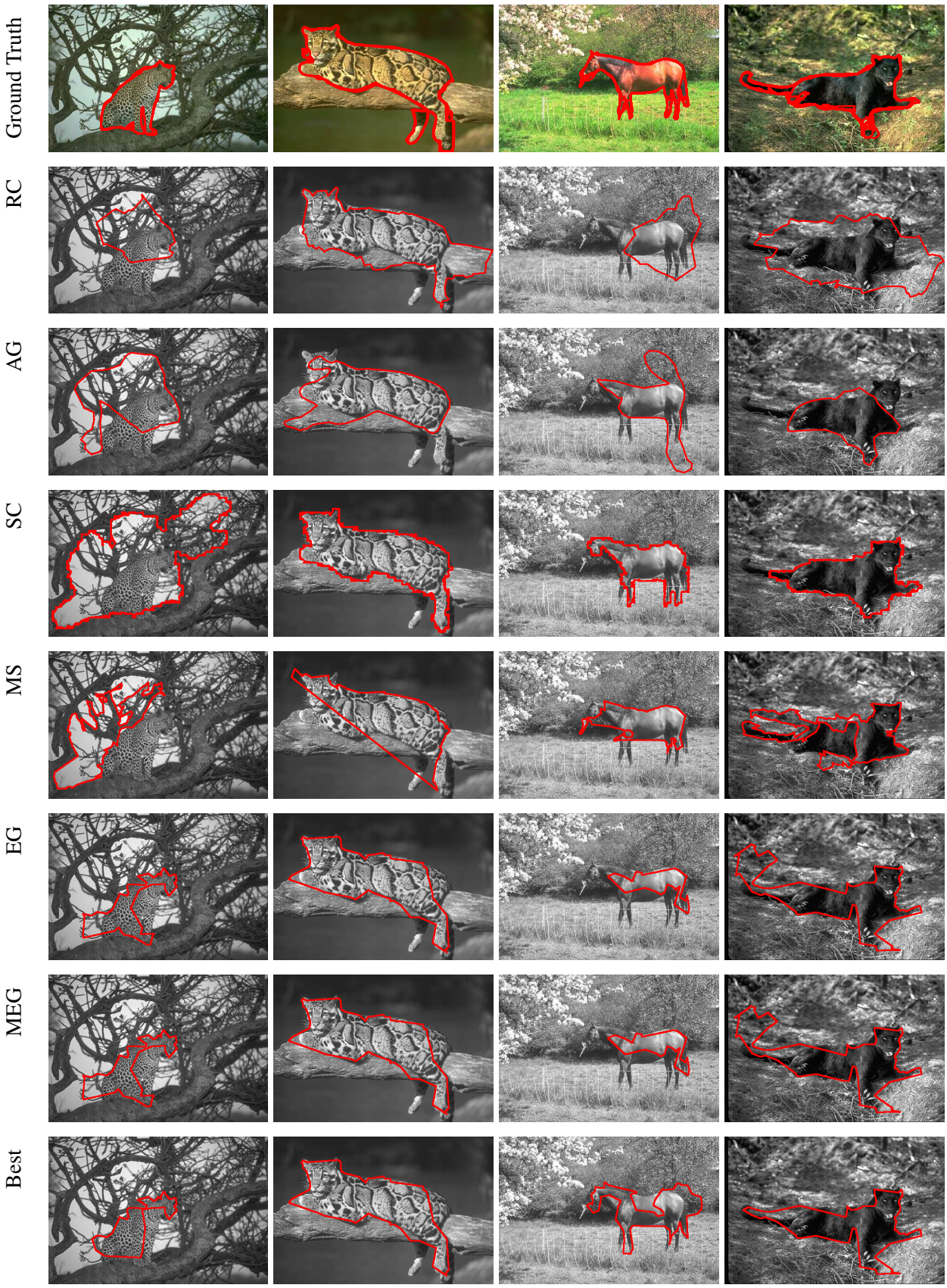



**Figure A.5: Qualitative results** - 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

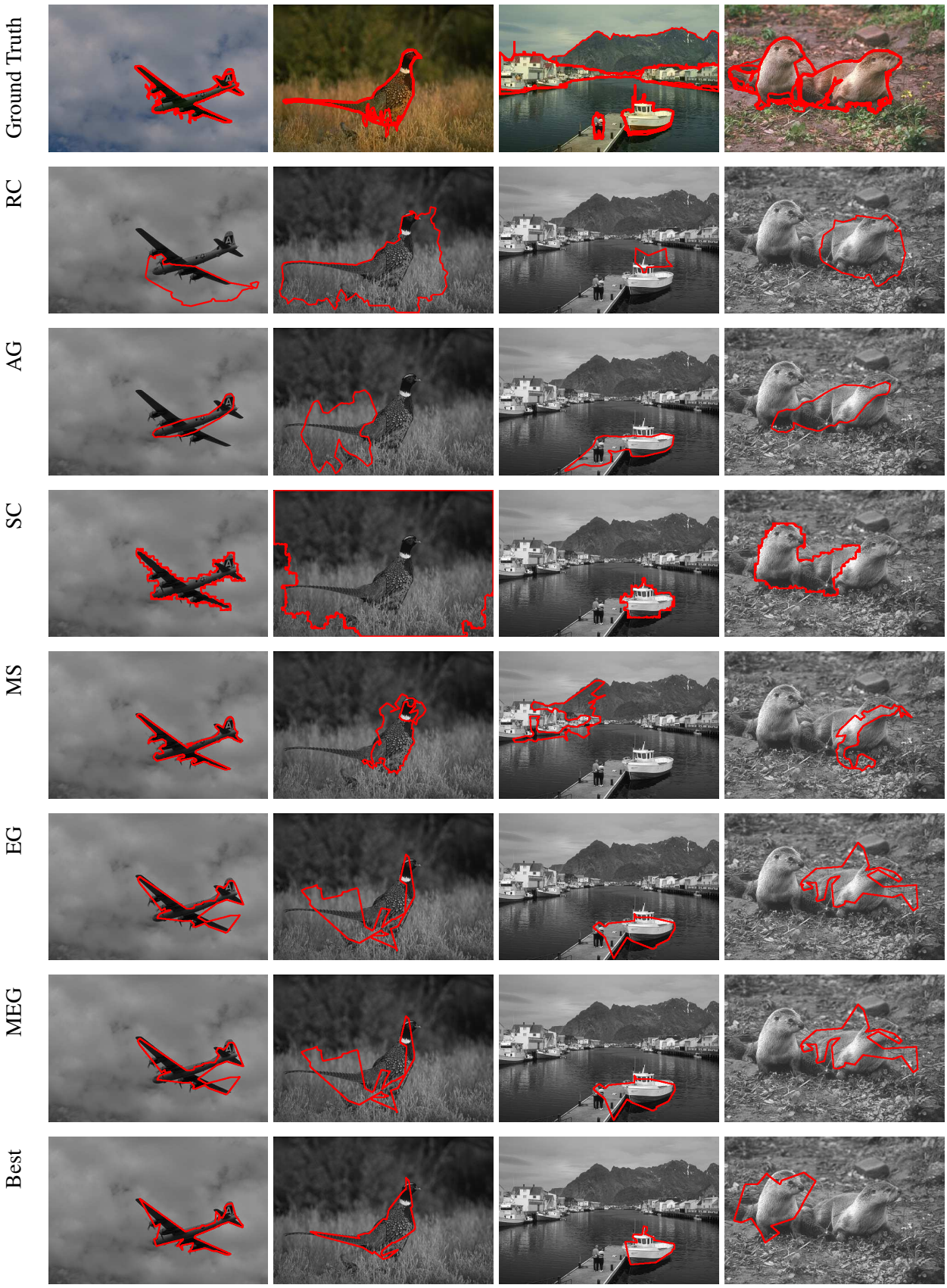



**Figure A.6: Qualitative results** - 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

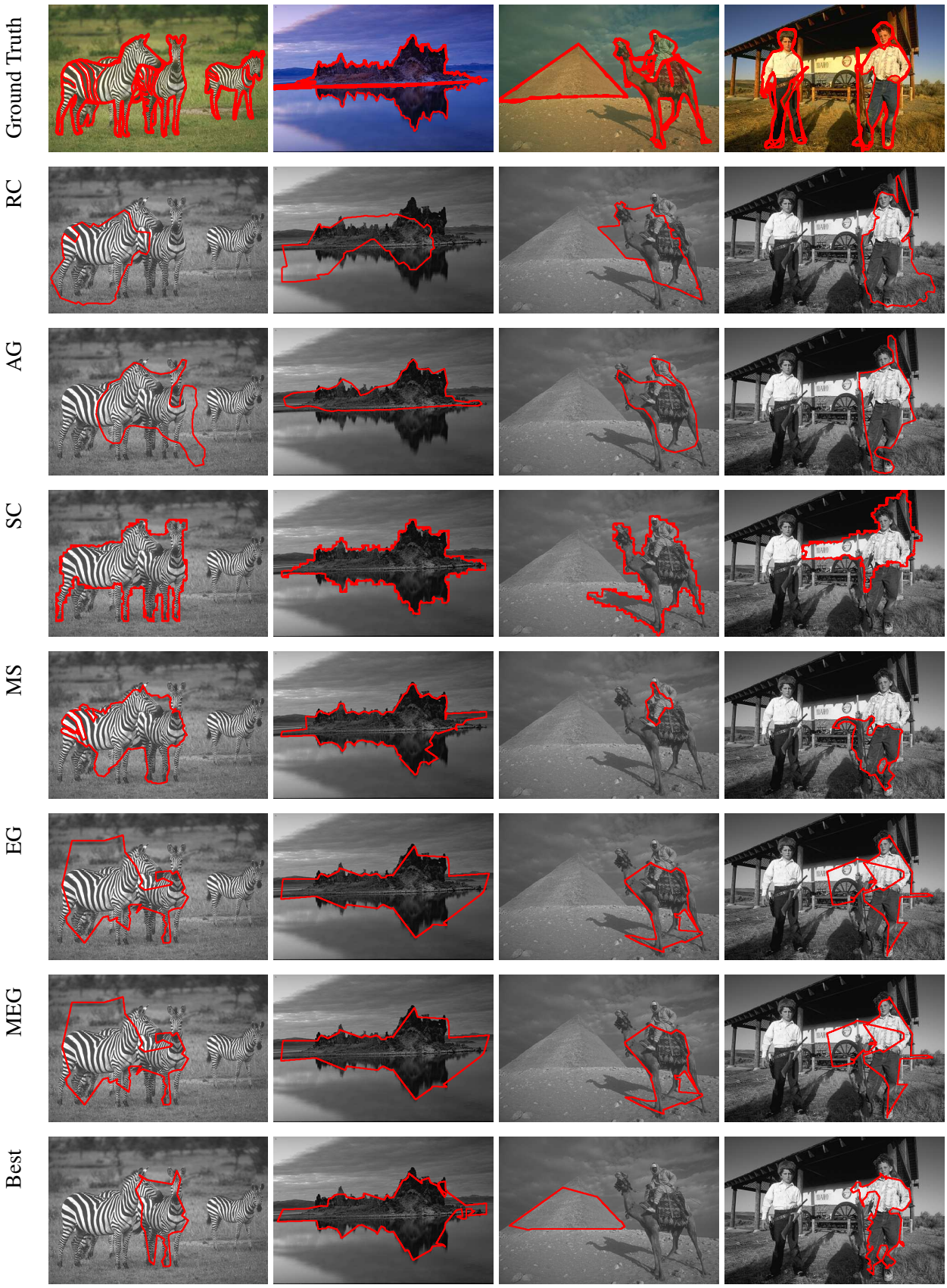



**Figure A.7: Qualitative results** - 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

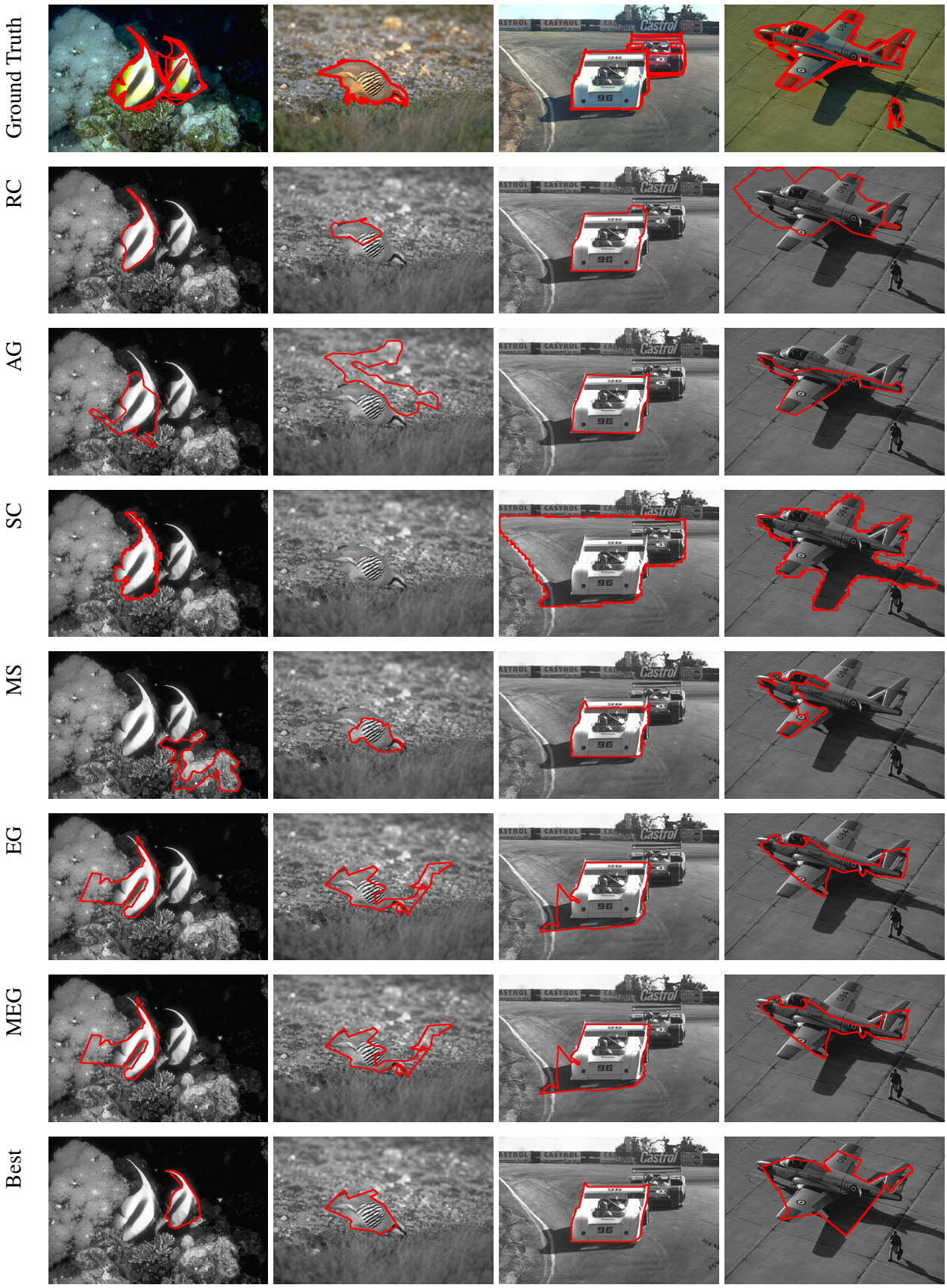



**Figure A.8: Qualitative results** - 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

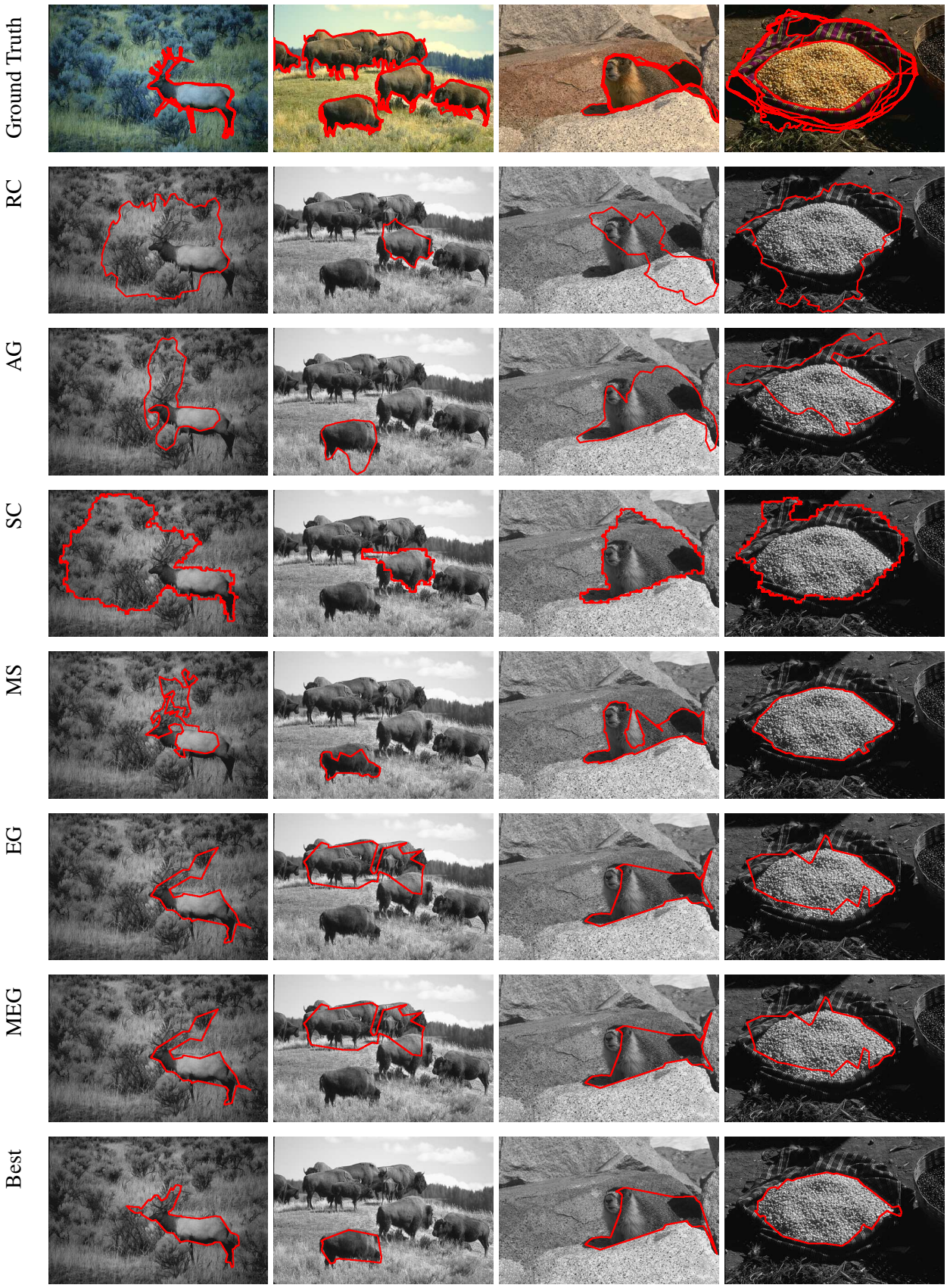



**Figure A.9: Qualitative results** - 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

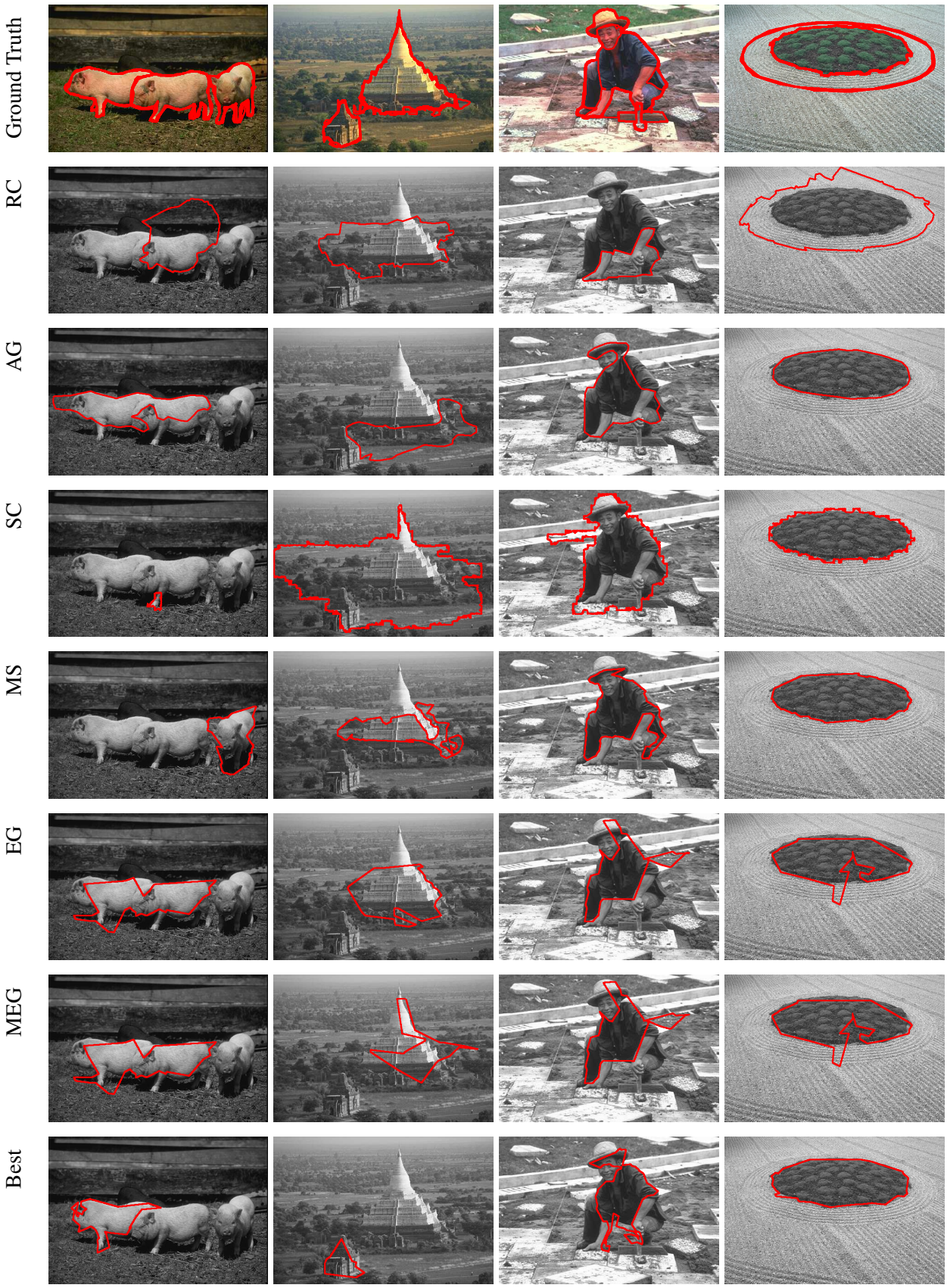



**Figure A.10: Qualitative results** - 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

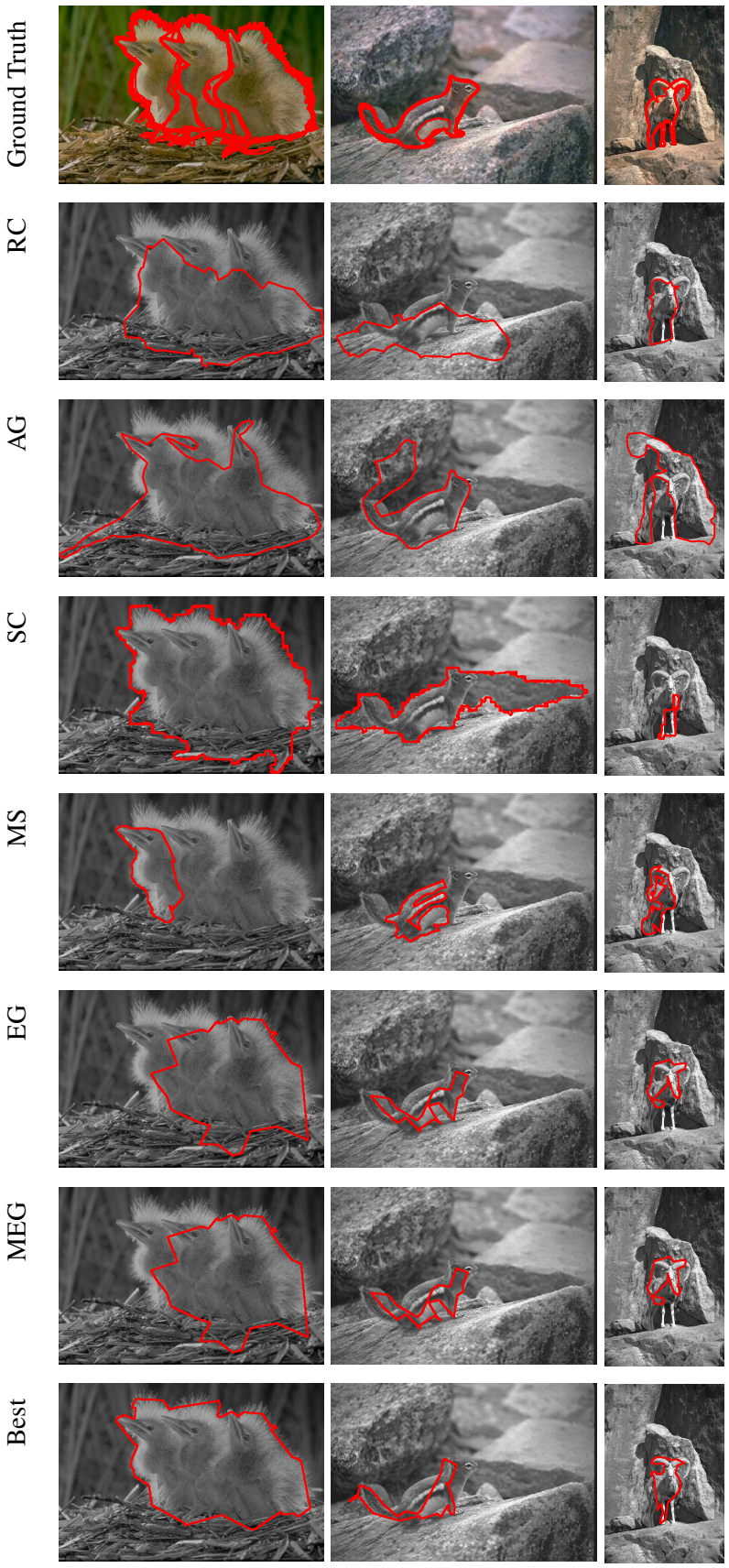



**Figure A.11: Qualitative results** - 1st row: Ground truth boundaries from **SOD**, 2nd row: best of top 20 of **RC**, 3rd row: best of top 20 of **AG**, 4th row: best of top 20 of **SC**, 5th row: best of top 20 of **MS**, 6th row: best of top 20 of **EG**, 7th row: best of top 20 of **MEG**, 8th row: best among all of MEG.

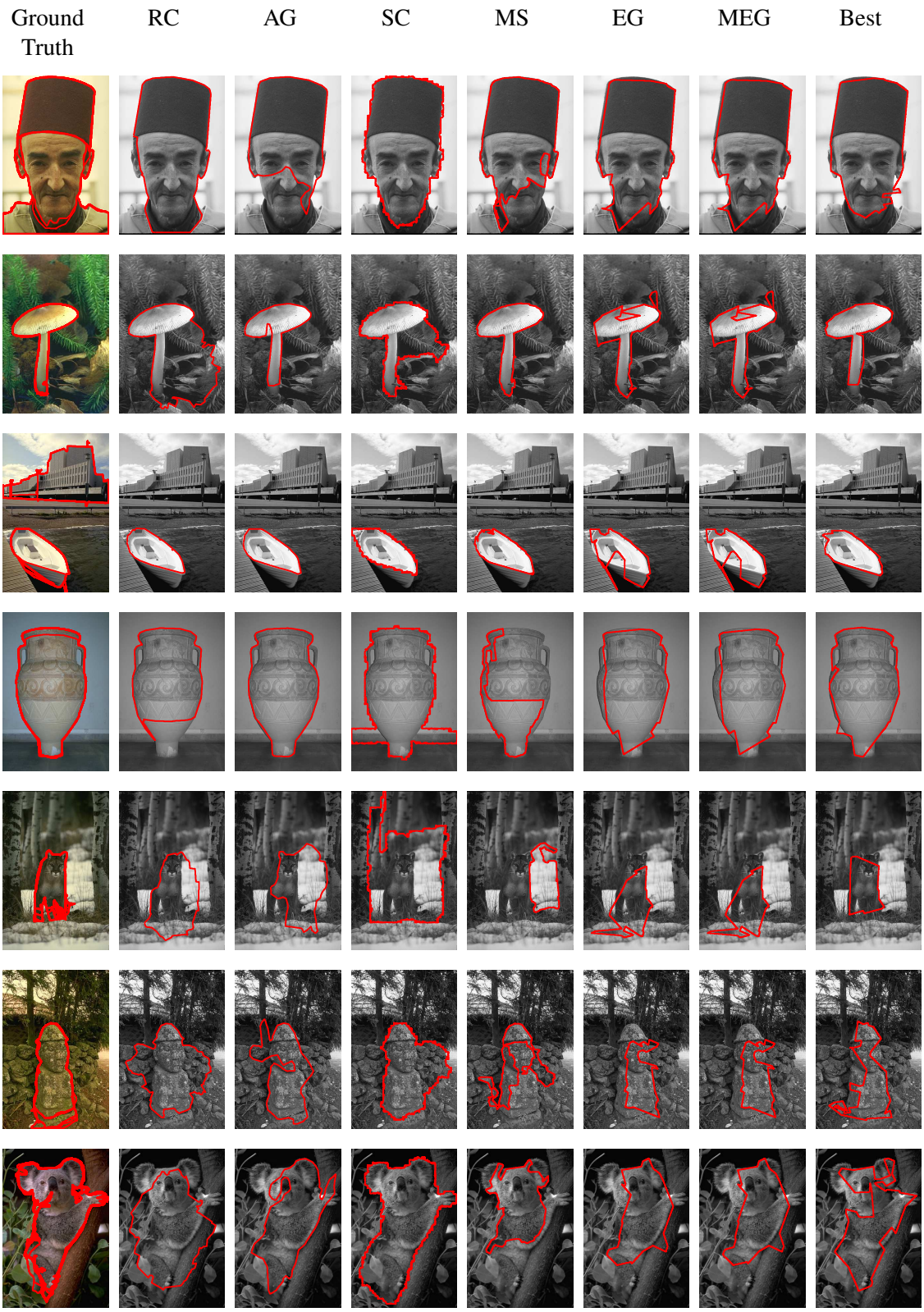



**Figure A.12: Qualitative results** - 1st column: Ground truth boundaries from **SOD**, 2nd column: best of top 20 of **RC**, 3rd column: best of top 20 of **AG**, 4th column: best of top 20 of **SC**, 5th column: best of top 20 of **MS**, 6th column: best of top 20 of **EG**, 7th column: best of top 20 of **MEG**, 8th column: best among all of MEG.

# References


[1] F. GE, S. WANG, AND T. LIU. **Image-Segmentation Evaluation From the Perspective of Salient Object Extraction**. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, **1**:1146–1153, 2006. 1, 37, 39

[2] GUANGCAN LIU, ZHOUCHEN LIN, XIAOOU TANG, AND YONG YU. **A hybrid graph model for unsupervised object segmentation**. *IEEE 11th International Conference on Computer Vision (ICCV 2007)*, pages 1–8, 2007. 1

[3] D. MARTIN, C. FOWLKES, D. TAL, AND J. MALIK. **A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics**. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, **2**, pages 416–423, 2001. 2, 4, 6, 7, 21, 30, 36, 37, 39, 62, 63, 64, 133

[4] XIANGHUA XIE AND MAJID MIRMEHDI. **Initialisation-Free Active Contour Segmentation**. In *20th International Conference on Pattern Recognition (ICPR 2010)*, pages 23 – 26, August 2010. 2

[5] J. H. ELDER AND R. M. GOLDBERG. **Ecological statistics of Gestalt laws for the perceptual organization of contours**. *Journal of Vision*, **2**[4]:324–353, 2002. 2, 5, 21, 23, 71, 72, 74, 75, 76, 77

[6] J. H. ELDER. **Are edges incomplete?** *International Journal of Computer Vision*, **34**[2-3]:97–122, AUG 1999. 2, 61, 62

[7] M. WERTHEIMER. **Untersuchungen zur Lehre von der Gestalt. II**. *Psycologische Forschung*, **4**[1]:301–350, 1923. 3, 11

[8] X. REN, C. C. FOWLKES, AND J. MALIK. **Learning probabilistic models for contour completion in natural images**. *International Journal of Computer Vision*, **77**[1-3]:47–63, 2008. 5, 21, 23

[9] J. H. ELDER, A. KRUPNIK, AND L. A. JOHNSTON. **Contour grouping with prior models**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**[6]:661–674, JUN 2003. 5, 21, 22, 23, 24, 25, 26, 27, 37, 71, 72, 89, 90, 91, 117

[10] F. J. ESTRADA AND J. H. ELDER. **Multi-scale contour extraction based on natural image statistics**. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06*, page 183, 2006. 5, 21, 22, 24, 28, 30, 69, 71, 72, 86, 88, 90, 91, 100, 106, 117, 118, 119, 133, 135, 144

[11] J. S. STAHL, K. OLIVER, AND S. WANG. **Open boundary capable edge grouping with feature maps**. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPR Workshops 2008. IEEE Computer Society Conference on*, pages 1–8, 2008. 5, 11, 17, 105, 111

[12] Z. WU AND R. LEAHY. **An optimal graph theoretic approach to data clustering: theory and its application to image segmentation**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**[11]:1101 – 1113, 1993. 9

[13] A. KELEMEN, G. SZEKELY, AND G. GERIG. **Elastic model-based segmentation of 3-D neuroradiological data sets**. *IEEE Transactions on Medical Imaging*, **18**[10]:828 – 839, 1999. 9, 10

[14] LI HE, HUI WANG, AND HONG ZHANG. **Object detection by parts using appearance, structural and shape features**. In *Mechatronics and Automation (ICMA), 2011 International Conference on*, pages 489–494, Aug 2011. 10

[15] YUN-TING LIN, YEN-KUANG CHEN, AND S-Y KUNG. **Object-based scene segmentation combining motion and image cues**. In *Image Processing, 1996. Proceedings., International Conference on*, **1**, pages 957–960 vol.1, Sep 1996. 10

[16] RAN SHI, ZHI LIU, YINZHU XUE, AND XIANG ZHANG. **Interactive object segmentation using iterative adjustable graph cut**. In *Visual Communications and Image Processing (VCIP), 2011 IEEE*, pages 1–4, Nov 2011. 10

[17] S. WANG, T. KUBOTA, J. M. SISKIND, AND J. WANG. **Salient closed boundary extraction with ratio contour**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**[4]:546–561, 2005. 10, 13, 14, 16, 17, 89, 102, 105, 111

[18] J. H. ELDER AND R. M. GOLDBERG. **Image editing in the contour domain**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **23**[3]:291–296, MAR 2001. 10

[19] J. S. STAHL AND S. WANG. **Edge grouping combining boundary and region information**. *IEEE Transactions on Image Processing*, **16**[10]:2590–2606, 2007. 11, 13, 14, 17, 18, 31, 33, 105, 111, 135

[20] A. LEVINSHTEIN, C. SMINCHISESCU, AND S. DICKINSON. **Optimal contour closure by superpixel grouping**. *Proceedings of European Conference on Computer Vision*, 2010. 11, 31, 32, 33, 104, 105, 111, 135

[21] J. S. STAHL AND S. WANG. **Globally optimal grouping for symmetric closed boundaries by combining boundary and region information**. *IEEE Transac-*

*tions on Pattern Analysis and Machine Intelligence*, **30**[3]:395–411, 2008. 11, 17, 91, 105, 111

[22] S. WANG, J. S. STAHL, A. BAILEY, AND M. DROPPS. **Global detection of salient convex boundaries**. *International Journal of Computer Vision*, **71**[3]:337–359, 2007. 11, 17, 91, 105, 111

[23] D. W. JACOBS. **Robust and efficient detection of salient convex groups**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**[1]:23–37, 1996. 13

[24] E. L. LAWLER. *Optimal cycles in doubly weighted linear graphs*, pages 209–214. Theory of Graphs. Dunod, Paris and Gordon and Beach, New York, 1967. 15

[25] R. KARP. **A characterization of the Minimum Cycle Mean in a Digraph**. *Discrete Math.*, **23**:309–311, 1978. 15

[26] J. EDMONDS. **Path, Trees, and Flowers**. *Canadian J. Math*, **17**:449–467, 1965. 16

[27] F. J. ESTRADA AND A. D. JEPSON. **Robust boundary detection with adaptive grouping**. In *Computer Vision and Pattern Recognition Workshop, CVPRW '06*, **2006**, pages 184–184, 2006. 17, 19, 20, 33, 91, 92, 102, 106, 107, 111, 135

[28] F. ATTNEAVE. **Some Informational Aspects of Visual Perception**. *Psychology Reviews*, **61**:183–193, 1954. 21

[29] H. B. BARLOW. *The coding of sensory messages*, pages 331–360. Current problems in animal behavior. Cambridge University Press, Cambridge, UK, 1961. 21

[30] E. BRUNSWIK AND J. KAMIYA. **Ecological cue-validity of proximity and of other Gestalt factors**. *American Journal of Psychology*, **66**:20–32, 1953. 21

[31] N. KRUGER. **Collinearity and parallelism are statistically significant second order relations of complex cell responses**. *Neural Processing Letters*, **8**:117–129, 1998. 21

[32] M. SIGMAN, G. A. CECHI, C. D. GILBERT, AND M. O. MAGNASCO. **On a common circle: Natural scenes and Gestalt rules**. *Proceedings of the National Academy of Sciences*, **98**:1935–1940, 2001. 21

[33] W. S. GEISLER, J. S. PERRY, B. J. SUPER, AND D. P. GALLOGLY. **Edge co-occurence in natural image predicts contour grouping performance**. *Vision Research*, **41**:711–724, 2001. 21

[34] P. FELZENSZWALB. **A min-cover approach for finding salient curves**. *Workshop on Perceptual Organization in Computer Vision*, **2006**:185, 2006. 21

[35] X. REN, C. C. FOWLKES, AND J. MALIK. **Scale-invariant contour completion using conditional random fields**. In *Proceedings of the 10th IEEE International Conference on Computer Vision*, **2**, pages 1214–1221, 2005. 21, 89

[36] D. R. MARTIN, C. C. FOWLKES, AND J. MALIK. **Learning to detect natural image boundaries using local brightness, color, and texture cues**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**[5]:530–549, MAY 2004. 24, 55, 56, 62

[37] J. H. ELDER AND S. W. ZUCKER. **Computing Contour Closure**. *Proc. 4th European Conference on Computer Vision*, **1**:399–412, 1996. 25, 89

[38] V. MOVAHEDI, F. J. ESTRADA, AND J. H. ELDER. **Global Cues for Multi-scale Probabilistic Grouping**. *Poster at Intelligent Systems Conference (IS08)*, 2008. 29

[39] ERAN BORENSTEIN AND SHIMON ULLMAN. **Class-Specific, Top-Down Segmentation**. In *Proceedings of the 7th European Conference on Computer Vision-Part II*, ECCV '02, pages 109–124, London, UK, UK, 2002. Springer-Verlag. 32, 33

[40] V. KOLMOGOROV, Y. BOYKOV, AND ROTHER C. **Applications of parametric maxflow in computer vision**. *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 32

[41] S. ALPERT, M. GALUN, R. BASRI, AND A. BRANDT. **Image segmentation by probabilistic bottom-up aggregation and cue integration**. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, 2007. 33, 37

[42] L. R. WILLIAMS. **Comparison of measures for detecting natural shapes in cluttered backgrounds**. *International Journal of Computer Vision*, **34**[2]:81, 1999. 33, 34

[43] S. MAHAMUD, L.R. WILLIAMS, K.K. THRONBER, AND K. XU. **Segmentation of multiple salient closed contours from real images**. *IEEE Transactions on Pattern analysis and Machine Intelligence*, **25**[4]:433, 2003. 34, 61, 71

[44] Q. ZOU, S. LUO, AND J. LI. **Selective attention guided perceptual grouping model**. In WANG L., CHEN K., AND ONG Y.S., editors, *First International Conference on Natural Computation, ICNC 2005*, **3610**, pages 867–876, 2005. 34

[45] J. J ZHONG, S. W LUO, AND Q. ZOU. **Perceptual grouping model for color natural images**. In *2007 International Conference on Wavelet Analysis and Pattern Recognition, ICWAPR '07*, **2**, pages 885–890, 2007. 34

[46] Q. ZHU, G. SONG, AND J. SHI. **Untangling Cycles for Contour Grouping**. *IEEE 11th International Conference on Computer Vision (ICCV 2007)*, pages 1–8, 2007. 34

[47] B. ALEXE, T. DESELAERS, AND V. FERRARI. **Measuring the Objectness of Image Windows**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**[11]:2189–2202, 2012. 34

[48] I. ENDRES AND D. HOIEM. **Category Independent Object Proposals**. *Proc. 11th European Conf. Computer Vision*, 2010. 34

[49] Y. J. ZHANG. **A survey on evaluation methods for image segmentation**. *Pattern Recognition*, **29**[8]:1335, 1996. 35

[50] V. MOVAHEDI AND J.H. ELDER. **Design and perceptual validation of performance measures for salient object segmentation**. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 49 –56, 2010. 36, 103, 134

[51] D. P. YOUNG AND J. M. FERRYMAN. **PETS Metrics: On-line performance evaluation service**. *Proceedings - 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, VS-PETS*, pages 317–324, 2005. 37, 39, 43, 145

[52] L. GOLDMANN, T. ADAMEK, AND P. VAJDA. **Towards fully automatic image segmentation evaluation**. *Lecture Notes in Computer Science*, **5259 LNCS**:566–577, 2008. 37, 39

[53] L. ZHOU, K. FU, Y. LI, Y. QIAO, X. HE, AND J. YANG. **Bayesian salient object detection based on saliency driven clustering**. *Signal Processing: Image Communication*, **29**[3]:434–447, 2014. 38

[54] L. ZHOU, Y. J. LI, Y. P. SONG, Y. QIAO, AND J. YANG. **Saliency driven clustering for salient object detection**. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 5372–5376, 2014. 38

[55] Z. LIU, W. ZOU, AND O. LE MEUR. **Saliency tree: A novel saliency detection framework**. *IEEE Transactions on Image Processing*, **23**[5]:1937–1952, 2014. 38

[56] QIAN HUANG AND B. DOM. **Quantitative methods of evaluating image segmentation**. In *Proceedings of the 1995 International Conference on Image Processing (ICIP'95)*, **3**, pages 53–56, 1995. 39

[57] F. C. MONTEIRO AND A. C. CAMPILHO. **Performance evaluation of image segmentation**. *Lecture Notes in Computer Science*, **4141 LNCS**:248–259, 2006. 39, 41, 94, 145

[58] DANIEL P. HUTTENLOCHER, G. A. KLANDERMAN, AND W. J. RUCKLIDGE. **Comparing images using the Hausdorff distance**. *IEEE Transactions on Pattern analysis and Machine Intelligence*, **15**[9]:850–863, 1993. 41

[59] D GEIGER, A GUPTA, L COSTA, AND VLONTZOS. **Dynamic programming for detecting, tracking and matching deformable contours**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**[3]:294–302, 1995. 43

[60] R. BASRI, L. COSTA, D.GEIGER, AND D. JACOBS. **Determining the Similarity of Deformable Shapes**. *Vision Research*, **38**:2365–2385, 1998. 43

[61] YORAM GDALYAHU AND DAPHNA WEINSHALL. **Flexible syntactic matching of curves and its application to automatic hierarchal classification of silhouettes**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**[12]:1313–1328, 1999. 43

[62] M. FRENKEL AND R. BASRI. **Curve Matching Using the Fast Matching Method**. *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 35–51, 2003. 43, 44

[63] CLAYTON SCOTT AND ROBERT NOWAK. **Robust Contour Matching via the Order-Preserving Assignment Problem**. In *IEEE Transactions on Image Processing*, **15:7**, pages 1831–1837, 2006. 43

[64] T. B. SEBASTIAN, P. N. KLEIN, AND B. B. KIMIA. **On aligning curves**. *IEEE Transactions on Pattern analysis and Machine Intelligence*, **25**[1]:116, 2003. 43

[65] F. R. SCHMIDT, D. FARIN, AND D. CREMERS. **Fast Matching of Planar Shapes in Sub-cubic Runtime**. In *Proc IEEE 11th Int Conf Computer Vision, ICCV 2007*, pages 1–6, 2007. 44

[66] MAURICE MAES. **On a cyclic string-to-string correction problem**. *Information processing letters*, **35**[2]:73, 1990. 44, 45, 47

[67] H. D. TAGARE, D. O'SHEA, AND D. GROISSER. **Non-rigid shape comparison of plane curves in images**. *Journal of mathematical imaging and vision*, **16**[1]:57, 2002. 44

[68] D. MUMFORD. **Mathematical theories of shape: do they model perception?** In *Proc SPIE Vol 1570 Geometric Methods in Computer Vision*, pages 2–10, 1991. 47, 145

[69] F. ESTRADA AND A.D. JEPSON. **Benchmarking image segmentation algorithms**. *International Journal of Computer Vision*, **85**:167–181, 2009. 56, 62

[70] V. MOVAHEDI AND J.H. ELDER. **Combining Local and Global Cues for Closed Contour Extraction**. In *24th British Machine Vision Conference (BMVC13)*, 2013. 62, 89, 94, 103

[71] J. CANNY. **Finding edges and lines in images**. Technical Report AITR-720, MIT Artificial Intelligence Laboratory, 1983. 62, 137

[72] PABLO ARBELAEZ, MICHAEL MAIRE, CHARLESS FOWLKES, AND JITENDRA MALIK. **Contour Detection and Hierarchical Image Segmentation**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**[5]:898–916, May 2011. 62, 64

[73] J. H. ELDER AND S. W. ZUCKER. **Local scale control for edge detection and blur estimation**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**[7]:699–716, 1998. 62, 137

[74] J. H. ELDER. **Scale space localization, blur, and contour-based image coding**. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 27, 1996. 66, 137

[75] CHRISTOPHER M. BISHOP. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 82

[76] H. HOTELLING. **Analysis of a complex of statistical variables into principal components**. *Journal of Educational Psychology*, **24**:417 – 441, 1933. 88

[77] D. B. JOHNSON. **Finding all the elementary circuits of a directed graph**. *SIAM J. Computing*, **4**[1]:77– 84, March 1975. 89, 90

[78] BEN ROBERTS AND DIRK P. KROESE. **Estimating the Number of s-t Paths in a Graph**. *Journal of Graph Algorithms and Applications*, **11**[1]:195–214, 2007. 90

[79] A. LEVINSHTEIN, C. SMINCHISESCU, AND S. DICKINSON. **Multiscale Symmetric Part Detection and Grouping**. *International Journal of Computer Vision*, pages 1–18, 2013. 91, 102, 105, 111

[80] L. T. ORU, I.AND MALONEY AND M. S. LANDY. **Weighted linear cue combination with possibly correlated error**. *Vision Research*, [43]:2451 – 2468, 2003. 95, 109

[81] F. J. ESTRADA AND A. D. JEPSON. **Perceptual grouping for contour extraction**. In *Proceedings of International Conference on Pattern Recognition*, **2**, pages 32–35, 2004. 106

[82] RADHAKRISHNA ACHANTA, SHEILA HEMAMI, FRANCISCO ESTRADA, AND SABINE SSSTRUNK. **Frequency-tuned Salient Region Detection**. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1597 – 1604, 2009. 107, 147

[83] P. CAVANAGH. *What's up in top-down processing?*, pages 295–304. Representations of Vision: Trends and Tacit Assumptions in Vision Research. Cambridge University Press, Cambridge, UK, 1991. 117

[84] P. D. KOVESI. **MATLAB and Octave Functions for Computer Vision and Image Processing**. Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>. 137

[85] A. JEPSON. **Robust Line Estimation**. University of Toronto. Available from: <http://www.cs.toronto.edu/∼fleet/courses/2503/fall11/Handouts/matlabVisTools.zip>. 137

[86] M. MAIRE. **Using contours to detect and localize junctions in natural images**. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pages 1 – 8, 2008. 137

[87] M. RIESENHUBER AND T. POGGIO. **Hierarchical models of object recognition in cortex**. *Nature Neuroscience*, **2**[11]:1019–1025, nov 1999. 145

[88] C.E. CONNOR, S.L. BRINCAT, AND A. PASUPATHY. **Transformation of shape information in the ventral pathway**. *Current Opinion in Neurobiology*, **17**[140-147], 2007. 145

[89] THORSTEN JOACHIMS. **Optimizing Search Engines Using Clickthrough Data**. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142, New York, NY, USA, 2002. ACM. 146

[90] D. SCULLEY. **Combined Regression and Ranking**. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 979–988, New York, NY, USA, 2010. ACM. 146

[91] L. ITTI, C. KOCH, AND E. NIEBUR. **A Model of Saliency-Based Visual Attention for Rapid Scene Analysis**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**[11]:1254 –1259, 1998. 147

[92] L ITTI AND C. KOCH. **Computational Modeling of Visual Attention**. *Nature Reviews Neuroscience*, **2**[3]:194 – 203, 2001. 147